

AD-A139 594

INTERACTIVE SIGNAL SYNTHESIS SYSTEM USER'S GUIDE(U)  
DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT  
CENTER BETHESDA MD I S ZARITSKY MAR 84 DTNSRDC-84/011

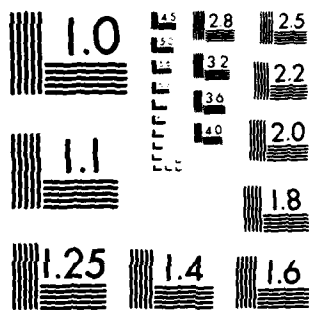
1/1

UNCLASSIFIED

F/G 9/2

NL

END  
DATE  
FILMED  
4 84  
DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS 1963-A

(12)

DTNSRDC-84/011

**DAVID W. TAYLOR NAVAL SHIP  
RESEARCH AND DEVELOPMENT CENTER**

Bethesda, Maryland 20084



AD A139594

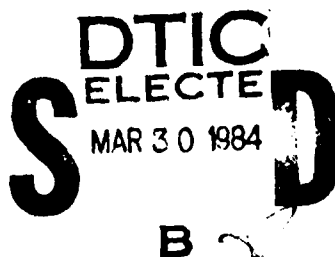
**INTERACTIVE SIGNAL SYNTHESIS SYSTEM  
USERS' GUIDE**

by

Irving S. Zaritsky

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT  
RESEARCH AND DEVELOPMENT REPORT



March 1984

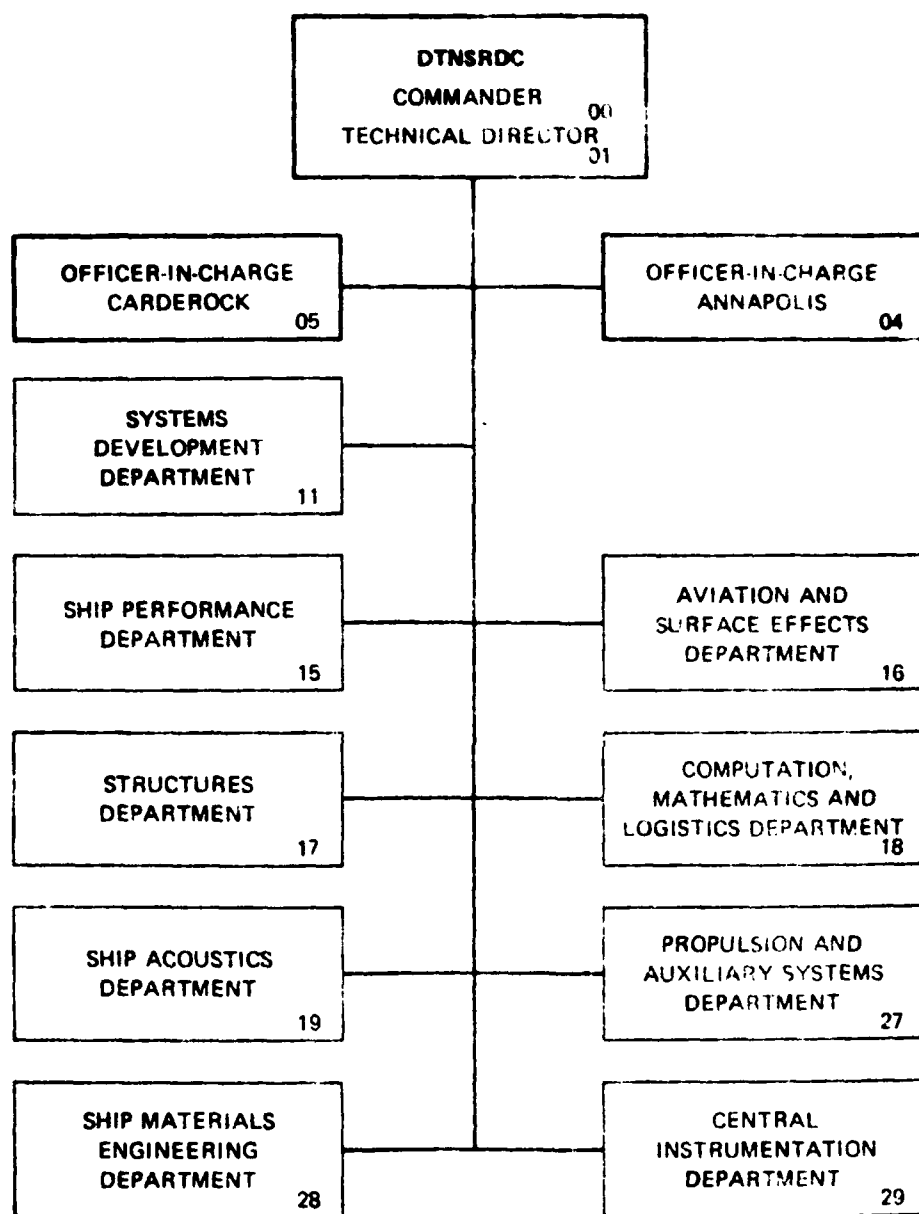
DTNSRDC-84/011

INTERACTIVE SIGNAL SYNTHESIS SYSTEM USERS' GUIDE

DTIC FILE COPY

84 03 30 012

# MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC-84/011 ✓	2. GOVT ACCESSION NO. AD-A139594	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  INTERACTIVE SIGNAL SYNTHESIS SYSTEM USERS' GUIDE	5. TYPE OF REPORT & PERIOD COVERED Final	
7. AUTHOR(s)  Irving S. Zaritsky	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS David Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084	8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command Washington, D.C. 20362	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS  (See reverse side)	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE March 1984	
	13. NUMBER OF PAGES 55	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)  APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Signal Synthesis                      Acoustic Transient Analysis Computer Science                      Transient Waveforms Acoustic Transients		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  The Interactive Signal Synthesis System is a user-oriented Fortran program designed to give the signal analyst the ability to generate controlled time series of arbitrary length and complexity. With the system, one can parameterize basic signal functions such as tone, gate, noise, polynomial, and arbitrary data, and also parameterize the shifted sum, damping, modulation, and filtering of each function. The decisions required of a user, when creating a function, are described.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE  
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Block 10)

Program Element 61153N

Task 15321 *17*

Task Area SR0140301

Work Unit 1808-010

**DTIC**  
**ELECTE**  
**S** MAR 30 1984 **D**  
**B**



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
<b>A-1</b>	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

## TABLE OF CONTENTS

	Page
LIST OF FIGURES AND TABLE . . . . .	iii
ABSTRACT . . . . .	1
ADMINISTRATIVE INFORMATION . . . . .	1
INTRODUCTION . . . . .	1
GETTING STARTED . . . . .	2
SUMMARY OF EXAMPLES . . . . .	7
EXAMPLE ONE - DMPSIN . . . . .	9
EXAMPLE TWO - POLNOM . . . . .	17
EXAMPLE THREE - SINMOD . . . . .	21
EXAMPLE FOUR - FAZMOD . . . . .	25
EXAMPLE FIVE - MODDAT . . . . .	31
EXAMPLE SIX - IMPTRN . . . . .	37
EXAMPLE SEVEN - IMPTRH . . . . .	44
EXAMPLE EIGHT - IMPTRL . . . . .	44
EXAMPLE NINE - NOISGT . . . . .	47
APPENDIX - DEFAULT EXTENSION NAMES . . . . .	49
REFERENCES . . . . .	51

### LIST OF FIGURES AND TABLE

1 - Time Series and Power Spectrum of DMPSIN . . . . .	18
2 - Time Series and Power Spectrum of POLNOM . . . . .	22
3 - Time Series and Power Spectrum of SINMOD . . . . .	26
4 - Time Series and Power Spectrum of SINE . . . . .	27
5 - Time Series and Power Spectrum of FAZMOD . . . . .	32
6 - Time Series and Power Spectrum of Original MODDAT . . . . .	38
7 - Time Series and Power Spectrum of RAMP . . . . .	39

	Page
8 - Time Series and Power Spectrum of MODDAT . . . . .	40
9 - Time Series and Power Spectrum of IMPTRN . . . . .	43
10 - Time Series and Power Spectrum of IMPTRH . . . . .	45
11 - Time Series and Power Spectrum of IMPTRL . . . . .	46
12 - Time Series and Power Spectrum of NOISGT . . . . .	48
<hr/>	
Table 1 - Summary of Examples . . . . .	8

## ABSTRACT

The Interactive Signal Synthesis System is a user-oriented Fortran program designed to give the signal analyst the ability to generate controlled time series of arbitrary length and complexity. With the system, one can parameterize basic signal functions such as tone, gate, noise, polynomial, and arbitrary data, and also parameterize the shifted sum, damping, modulation, and filtering of each function. The decisions required of a user, when creating a function, are described.

## ADMINISTRATIVE INFORMATION

The work reported here was performed in the Computer Science and Information Systems Division of the Computation, Mathematics, and Logistics Department, within Program Element 61153N, Task Area SR0140301, Task 15321, and Work Unit 1808-010, the Mathematical Sciences Research Program under the sponsorship of NAVSEA 03R22.

## INTRODUCTION

The Interactive Signal Synthesis System is a user-oriented Fortran program designed to give the signal analyst the ability to generate controlled time series of arbitrary length and complexity. With the system, one can parameterize basic signal functions such as tone, gate, noise, polynomial, and arbitrary data, and also parameterize the shifted sum, damping, modulation, and filtering of each function.

For the convenience of the non-computer-oriented analyst, emphasis has been placed on structuring the interactive user interface with prompts, definitions, and optional echoes. However, for exact formula definitions and discussions of side effects, the user must refer to the material which follows. The user is also referred to Berkowitz<sup>1\*</sup> who describes the algorithmic aspects of the system. Output is formatted and scaled for viewing and manipulation by ISPARS (Interactive Signal Pattern Analysis and Recognition System).<sup>2</sup> Although the system has been coded specifically in Fortran for the PDP-11/45, it contains no graphics or assembly language code; its portability therefore is strictly a function of its machine-dependent I/O.

---

\*A complete listing of references is given on page 51.

It is expected that the signal synthesis system will be of use in generating test data for validation of signal analysis programs, in establishing an analysis-by-synthesis procedure for pattern classification application, and in determining standards for transient waveform definition.

This report describes the decisions required of a user in executing the program.

### GETTING STARTED

In order to best show both the capabilities of the synthesizer and the decisions that must be made by a user, a series of interactive sessions will be described. The output from these sessions will be displayed using an X-Y plotter which is integrated into ISPARS. ISPARS will also be used to compute and display a power spectrum.

Every time the program is executed, an ASCII file containing all the parameters necessary to concisely describe the signal is created. This file will be referred to as the parameter file and a listing of each one will be included with each session description. This file may be used to recreate and, if desired, to modify the signal at a later time. Note that the parameter file may be easily modified by the RT-11 system editor (EDIT) and also concatenated with other parameter files by using the RT-11 peripheral interchange program (PIP).

Both prompts and responses by the computer will be given in capital letters. User responses will be underlined and descriptions will be given in lower case lettering. It is assumed that the user has only limited experience with the computer.

There are two versions of the synthesizer: one designed for the PDP-11/45 and the other for the PDP 11/40. Both versions are called SIGNAL.

To begin an interactive session with the synthesizer, enter

.R SIGNAL

Of course, if the synthesizer is on a peripheral device other than the system disk, the user should instead enter:

.RUN dev:SIGNAL

Whenever the program returns with

PAUSE--"statement",

the user must respond by entering a carriage return, represented by <CR>.

During the course of the program, temporary scratch files are created. Depending on the version of SIGNAL chosen, the user will see one of the following prompts:

TURN RK1: DISK DRIVE ON  
DO NOT PLACE OUTPUT FILES ON RK1:  
PAUSE--HIT <CR> TO CONTINUE

or

TURN RJPO4 DISK DRIVE ON  
DO NOT PLACE OUTPUT FILES ON DP2:  
PAUSE--HIT <CR> TO CONTINUE

Since only integer files can be operated on by ISPARS, we answer yes to the next question.

DO YOU WISH TO CREATE AN INTEGER OUTPUT FILE ?  
1=YES, 0=NO  
1

A negative response to this question and the next one will result in creating only a parameter file.

DO YOUR WISH TO CREATE A REAL OUTPUT FILE ?  
1=YES, 0=NO  
0

Note that, while a file containing real values cannot presently be used with ISPARS, such a file has the advantage of higher precision and may be useful since the synthesizer accepts both real and integer files as a DATA function input.

The user is then asked whether or not this session will be interactive.

DO YOU WISH TO INTERACT WITH THE PROGRAM ?  
1=YES, 0=NO  
1

The program allows for three degrees of interaction:

- 1) Create a new signal with input from a terminal  
(User response = 1)
- 2) Use a previously created parameter file and modify it  
(user response = 1) Typically, modifications may include:
  - a) Modifying a parameter value  
For instance, changing a tone frequency from 1 khz to 2 khz
  - b) Switching function types  
For instance, TONE to GATE or EXIT to NOISE

c) Changing modulation type

For instance, AMPLITUDE to PHASE, (spectral) FREQUENCY to NONE, or to modulate a signal which was originally unmodulated

d) Changing filter mode

For instance, filter a previously unfiltered signal, or vice versa

e) Changing filter and window type

For instance, high pass to low pass filter and Hamming to Chebyshev window

- 3) Use a previously created parameter file in its original form  
(User response = 0)

This mode of operation represents a time-space trade-off between storing the parameter file, which rarely exceeds one disk block (256 words) and storing the digitized signal which, for  $n$  points, takes  $1 + [1 + ((n-1)/256)]$  disk blocks where  $[ ]$  stands for "greatest integer in." This value includes space for a header block, as required by ISPARS, which contains the sampling rate and the number of channels (=1).

A positive response to the next question

DO YOU WANT INSTRUCTION / VERIFICATION?

1=YES, 0=NO

1

will have two effects:

- 1) Many of the prompts throughout the session will have an expanded description.
- 2) The user can immediately accept or reject the most recently entered value.

The system then prompts the user (with an asterisk) to assign names to the input and output files.

ENTER OUTPUT AND INPUT PARAMETER FILE NAMES

IN COMMAND STRING FORMAT:

OUTPUT[.EXT]=INPUT[.EXT]

NOTE:

TO INPUT PARAMETERS EXCLUSIVELY FROM THE TERMINAL,  
SET INPUT PARAMETER FILE NAME TO:

"TT:"

.PAR IS DEFAULT EXTENSION

\*DMPSIN=TT:

The names may have up to six characters and the optional extension may have up to three characters. If the extension is left off, a default extension ".PAR" is automatically included:

NEWFIL=OLDFIL

becomes: NEWFIL.PAR=OLDFIL.PAR.

All RT-11 files may have both a name and an extension to further describe and locate these files. The synthesizer was written to allow all pertinent files to have default names, thus saving the user both time and effort. These extension names and their associated file descriptions are listed in the Appendix. Note that a user may enter

OLDFIL=OLDFIL

However, the user should be aware that this entry will destroy the original parameter file.

There are some further points of interest:

- . A user may prefer to place the new file somewhere other than on the system disk. This is accomplished by adding dev:NEWFIL where "dev" may be set to RK1, DPO, MT1 etc. depending on the desired physical location.\*
- . If the user is creating a new parameter file, and thus entering all information via the terminal, the input file name should be set to "TT:"
- . The synthesizer will check the validity of the input parameter file name. A misspelled or nonexistent file name will bring the following message:

INPUT FILE DOES NOT EXIST, TRY AGAIN !  
DO YOU WISH TO INTERACT WITH THE PROGRAM ?  
1=YES, 0=NO

.  
.  
.

The program resequences at this point.

The synthesizer then asks for output file names. The integer signal file has a default extension of ".SIG" and the floating point signal file default extension is ".WAV".

---

\* See the PDP-11 RT-11 System Reference Manual<sup>3</sup> for details.

ENTER NAME OF THE INTEGER OUTPUT SIGNAL FILE: "FILE[.EXT]="

NOTE:

DEFAULT EXT IS: .SIG

\*DMPSIN=

If the user had previously requested creation of the real output file, the following prompt would have been issued:

ENTER NAME OF REAL OUTPUT SIGNAL FILE: "FILE[.EXT]="

NOTE:

DEFAULT EXT IS: .WAV

\*DMPSIN=

The user must include the equal sign after the file name as this instructs the RT-11 SYSLIB function ICSI that the file name is for output. The synthesizer checks for the equal sign and an incorrect entry will produce the following message:

ENTER NAME OF THE INTEGER OUTPUT SIGNAL FILE: "FILE[.EXT]="

\*DMPSIN

PAUSE -- CHECK FOR "="

<CR>

ENTER NAME OF THE INTEGER OUTPUT SIGNAL FILE: "FILE[.EXT]="

\*DMPSIN=

The user is then prompted for the sampling rate, which has been chosen here to be 18000 hz. That rate will be used for all examples except the second. The user must precede the value with either an "F" for frequency or a "T" for sampling period. If this indication is omitted, the following message will appear:

ENTER EITHER THE SAMPLING FREQ. ("F"G15.8)  
OR THE SAMPLING PERIOD ("T"G15.8)

18000.0

PAUSE -- ERROR, BAD FREQ. OR PERIOD VALUE

<CR>

ENTER EITHER THE SAMPLING FREQ. ("F"G15.8)  
OR THE SAMPLING PERIOD ("T"G15.8)

F18000.

SAMPLING FREQUENCY IS 18000.000  
SAMPLING PERIOD IS 0.55555556E-04  
1=YES, 0=NO

1

#### SUMMARY OF EXAMPLES

The nine examples are listed and briefly described in Table 1. Included in this description will be the file name, which is the same for both the parameter and output files (different extensions, of course), location of the input parameters, modulation type, and whether or not the following features were requested:

- . Delayed repeat sum
- . Function damping
- . Filtering
- . Instruction/verification

Integer signals have been scaled to + or - 2047. Also, for best resolution on the X-Y plotter, the following parameters are set:\*

- . Vertical display scale = 2.0
- . L1 = 30
- . Z1 = 40
- . Power spectra uses trapezoidal window
- . 20 points/side

---

\*See ISPARS<sup>2</sup> manual for a description of these parameters.

TABLE 1 - SUMMARY OF EXAMPLES

File Name	Function(s)*	Description										Parameter File Input			Instruction Verification
		Delayed Repeat Sum**	Damping***	Type	Modulation Function	Filtering		Damping	Interactive Input from:	Non-Interactive	Tele- Old File				
						Individual Function	Summed Function								
1. DMP5IN	a. Tone b. Noise (uniform) c. Summation is filtered	Y; 180 N	Y; 60.0 N	None None			N N	Y; Hi Pass				Y			
2. POLXOM	Polynomial	N	N	None			N	N				N			
3. SINMOD	Tone	N	N	Amplitude	Tone	N	N	N				Y			
4. FAZMOD	Tone	N	N	Phase	Tone	N	N	N				N			
5. MODDAT	Data (Input from SINE, SIG)	N	N	Amplitude	Polynomial: (Ramp)	N	N	N				Y			
6. IMPTRN	Gate (Impulse Train)	N	N	N			N	N							
7. IMPTRH	Gate (Filtered Impulse Train)	N	N	N			Y; Hi Pass	N							
8. IMPTRL	Gate (Filtered Impulse Train)	N	N	N			Y; Low Pass	N							
9. NOISGT	a. Gate (50% Duty Cycle) b. Gaussian Noise	N	N	N			N	N							

\*If example has multiple functions, they are added.  
\*\*If yes, the number of occurrences is listed, along with the delay (in points).  
\*\*\*If yes, z (alpha) is also listed.  
If yes, the filter type is listed.

\*If example has multiple functions, they are added.

\*\*If yes, the number of occurrences is listed, along with the delay (in points).

\*\*\*If yes,  $\alpha$  (alpha) is also listed.

† If yes, the filter type is listed.

# EXAMPLE ONE - DMPSIN

The program flow resumes with the program beginning its major cycle with a request for a function type.

ENTER FUNCTION TYPE:  
EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

TONE

FUNCTION TYPE IS: "TONE"

The user may respond with the full name, such as "TONE", or with an abbreviation, e.g., "T". The same rule holds for any of the other function names. Note that the user must not place any blanks before the first character as this would be an error and result in the message:

PAUSE -- ERROR, INCORRECT FUNCTION TYPE  
<CR>

Since "INSTRUCTION/VERIFICATION" was requested, after a correct entry the program will also respond with

1=YES, 0=NO  
1

thus giving the user a chance to make a change.

One of the features of the signal synthesizer is the ability to take a signal, delay it in "time", and then add it to the original signal. This feature will be referred to as a "delayed repeat sum." The user may choose up to 32766 repeats of the original signal. The final signal may contain as many samples as the disk will hold. The interactive session continues:

ENTER # OF SAMPLES FOR FIRST OCCURRENCE (F10.0 <CR>  
AND # OF OCCURRENCES I5 <CR>  
1025  
2

# OF SAMPLES IS 1025.  
# OF OCCURRENCES IS 2  
1=YES, 0=NO  
If, at this point, the user wishes to make a change,  
0

ENTER # OF SAMPLES FOR FIRST OCCURRENCE (F10.0 <CR>  
AND # OF OCCURRENCES I5 <CR>  
1024  
2

# OF SAMPLES IS 1024.  
# OF OCCURRENCES IS 2  
1=YES, 0=NO

1

If the user indicates only one occurrence of the signal, the following request is omitted:

ENTER THE OCC. FREQ. ("F"G15.8)  
OR THE OCC. PERIOD ("T"G15.8)  
OR THE DELAY [IN PTS] ("D"G15.8)

F100.0

OCCURRENCE FREQ. IS 100.00000  
OCCURRENCE PERIOD IS 0.9999998E-02  
DELAY (SAMP FRQ/OCC FRQ) IS 180.00000

1=YES, 0=NO

1

The ratio of occurrence period/sampling period determines the time delay in samples. Therefore, in order to have multiple occurrences properly aligned, this ratio must be a non-negative integer. Consequently, the occurrence frequency or period should be carefully chosen. If this ratio is not a non-negative integer, the following warning will be issued:

ENTER THE OCC. FREQ. ("F"G15.8)  
OR THE OCC. PERIOD ("T"G15.8)  
OR THE DELAY [IN PTS] ("D"G15.8)

F101.0

OCCURRENCE FREQ. IS 101.00000  
OCCURRENCE PERIOD IS 0.99009899E-02  
DELAY (SAMP FRQ/OCC FRQ) IS 178.21782

1=YES, 0=NO

1

\*\*\* WARNING \*\*\*  
\*\*\*, DELAY SHOULD BE A NON-NEGATIVE INTEGER  
2ND OCC WILL NOT ALIGN WITH FIRST. DELAY WAS ROUNDED OFF

TRY AGAIN ?

1=YES, 0=NO

1

ENTER EITHER THE SAMPLING FREQ. ("F"G15.8)  
OR THE SAMPLING PERIOD ("T"G15.8)

.

.

.

WITH REPETITION, TOTAL # OF SAMPLES WILL BE 1204.

When "instruction/verification" is on, the equation for the tone is listed, and then the individual parameters are requested.

$$\text{UNMOD. VAL} = A1 \sin(2\pi \cdot \text{FRQ} + \text{PH1}) + M1$$

ENTER EITHER THE TONE FREQUENCY ("F"G15.8) HZ  
OR THE TONE PERIOD ("T"G15.8)

F600.0

TONE FREQ IS 600.00000  
TONE PERIOD IS 0.16666667E-02  
2 \* PI \* FREQ IS 3769.9109  
1=YES, 0=NO

1

Note that when more than one parameter is entered on a line, the values may be entered separated by commas:

100.0,0.0,0.0

However, if the second and third values are zero, they may be left off entirely, as shown next.

ENTER TONE AMPLITUDE, MEAN,  
AND FRACTIONAL PHASE IN THE FORM:  
G15.8, G15.8, P.PPPPP <CR>

100.0

AMPLITUDE, MEAN, AND FRACTIONAL PHASE ARE:  
100.00000 0.00000000 0.00000000  
1=YES, 0=NO

1

Two contiguous commas denote a value of zero for the parameter in question. Also note that any non-zero values should include the decimal point. It was not included in the following entry, with unexpected results:

ENTER TONE AMPLITUDE, MEAN,  
AND FRACTIONAL PHASE IN THE FORM:  
G15.8, G15.8, P.PPPPP <CR>

100,,1.0

AMPLITUDE, MEAN, AND FRACTIONAL PHASE ARE:  
0.10000000E-05 0.00000000 1.00000000  
1=YES, 0=NO

The synthesizer automatically scales the final signal to a maximum absolute value of 2047.0 so that both the ISPARS graphics display and the X-Y plot of the time domain signal will be large enough to view without being cut off at the top. Therefore, the requested amplitude is needed only to determine the

relative amplitude of one signal to another. For instance, if another tone with an amplitude of 20.0 were added to this tone, its maximum amplitude would be 20% of the original tone.

The user should also be cautioned to enter a non-zero value (up to 1.0) for the phase when using phase modulation.

Next, the tone is damped at a rate of  $e^{-(\alpha * t)}$  where alpha has been set to 60.0 as follows.

```
ENTER EITHER THE TIME CONST.      ("T"G15.8)
      OR THE BANDWIDTH [TC/PI] ("F"G15.8)
      T60.0
```

```
TIME CONST. IS      60.000000
BANDWIDTH [TC/PI] IS  19.098595
1=YES, 0=NO
1
```

As indicated, the bandwidth is calculated to be the time constant divided by Pi.

The next request is for modulation type.

```
ENTER MODULATION TYPE:
AMPLITUDE, FREQUENCY (SPECTRAL), PHASE, NONE
NONE
```

```
MODULATION TYPE IS "NONE      "
1=YES, 0=NO
1
```

If a user had chosen one of the modulation types and also chosen a tone for the modulating function, the following information would have been returned:

```
AM VAL. =
A1 [A2 SIN(2 PI FRQ2 + PH2) + M2] SIN(2 PI FRQ1 + PH1) + M1
```

```
SPECTRAL FM VAL =
A1 SIN(2 PI FRQ1 [A2 SIN(2 PI FRQ2 + PH2) + M2] + PH1) + M1
```

```
PHASE MOD VAL =
A1 SIN(2 PI FRQ1 [A2 SIN(2 PI FRQ2 + PH2) + M2] + PH1) + M1
```

where A1, FRQ1, PH1, and M1 are the carrier tone parameters and A2, FRQ2, PH2, and M2 are the modulating tone parameters.

The user then has an opportunity to filter this individual signal before it is added to the sum file.

```
DO YOU WISH TO FILTER THIS SIGNAL ?
1=YES, 0=NO
0
```

At this point, the signal is computed and added to the sum file. Since this is the first function created, the sum file contains only the signal just created.

DO YOU WISH TO FILTER THE SUMMED SIGNAL ?

1=YES, 0=NO

0

We will now add uniform noise to the original signal:

ENTER FUNCTION TYPE:

EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

NOISE

FUNCTION TYPE IS: "NOISE"

1=YES, 0=NO

1

ENTER # OF SAMPLES FOR FIRST OCCURRENCE (F10.0 <CR>  
AND # OF OCCURRENCES I5 <CR>)

1024

1

# OF SAMPLES IS 1024

# OF OCCURRENCES IS 1

1=YES, 0=NO

1

A misspelled or incorrect noise type will cause the following to be listed:

ENTER NOISE TYPE: (UNIFORM OR GAUSSIAN) <CR>  
RANDOM

PAUSE -- INCORRECT NOISE TYPE, TRY AGAIN !  
<CR>

ENTER NOISE TYPE: (UNIFORM OR GAUSSIAN) <CR>  
UNIFORM

NOISE TYPE IS "UNIFORM "

1=YES, 0=NO

1

For the convenience of the reader, equation (20) from Berkowitz<sup>1</sup> is repeated here:

$$d_{4u}(t) = A * SIGMA * e^{-(ALPHA * t)} * SQRT(12) * (RAN(t) - .5) + MEAN$$

The user is then asked for A, MEAN, and SIGMA.

ENTER AMPLITUDE, MEAN AND STD DEV: (3G15.8 <CR>)

20.,,1.

AMPLITUDE,            MEAN AND STD DEV ARE:  
20.000000            0.00000000            1.00000000  
1=yes, 0=no  
1

The time constant is ALPHA.

ENTER EITHER THE TIME CONST.            ("T"G15.8)  
OR THE BANDWIDTH [TC/PI]            ("F"G15.8)  
TO.0

TIME CONST. IS            0.00000000  
BANDWIDTH [TC/PI] IS    0.00000000  
1=YES, 0=NO  
1

Here, the user is asked for a sequence of four integers. The range is 0 to 32767. These numbers are used by the PDP random number generator RAN() and the use of the same four integers will always result in the same sequence generated by RAN().

FOR RANDOM NUMBER GENERATION,  
ENTER TWO PAIRS OF INTEGERS IN THE FORM: N1, N2, N3, N4 <CR>  
327,964,31855,2

N1,        N2,        N3,        N4 ARE:  
327        964        31855        2  
1=YES, 0=NO  
1

ENTER MODULATION TYPE: AMPLITUDE OR NONE  
NONE

MODULATION TYPE IS "NONE        "  
1=YES, 0=NO  
1

At this point, the user may wish to filter the noise.

DO YOU WISH TO FILTER THIS SIGNAL ?  
1=YES, 0=NO  
0

The noise signal is now created and added to the original signal.

DO YOU WISH TO FILTER THE SUMMED SIGNAL ?  
1=YES, 0=NO  
1

The user may have up to 30 band-pass, band-stop, hi-pass, and low-pass filter bands, each with its own gain setting.

ENTER # OF FILTERBANDS, INCLUDING HIGH AND LOW PASS  
LIMIT IS 30

1

# OF FILTERBANDS IS: 1

1=YES, 0=NO

1

ENTER 1 GAIN VALUES: (5G15.8 FORMAT)

1.0

GAIN VALUES ARE:

1.00000

1=YES, 0=NO

1

\*\*\* WARNING, DO NOT NORMALIZE INPUTS \*\*\*

THE FILTER TYPES ARE:

1 - LOW PASS

2 - HIGH PASS (FILTER LENGTH MUST BE ODD)

3 - BAND PASS (FILTER LENGTH MUST BE ODD)

4 - BAND STOP

THE WINDOW TYPES ARE:

1 - RECTANGULAR

2 - TRIANGULAR

3 - HAMMING

4 - GENERALIZED HAMMING (REQ. ALPHA)

5 - HANNING

6 - KAISER (REQ. STOPBAND ATTENUATION TO COMPUTE BETA)

7 - CHEBYSHEV (REQ RIPPLE (DB) AND/OR TRANSITION WIDTH)

The number of filter weights (filter length) may not exceed 512.

SPECIFY FILTER TYPE(I2), WINDOW TYPE(I2), FILTER LENGTH(I4)

2,5,255

FILTER TYPE, WINDOW TYPE, FILTER LENGTH ARE: 1 5 255

1=YES, 0=NO

1

SPECIFY IDEAL CUTOFF FREQUENCY(G15.8)

300.0

IDEAL CUTOFF FREQUENCY IS 300.00000 HZ

1=YES, 0=NO

1

A high-pass filter was chosen. If a band-pass or band-stop filter were used, the synthesizer would have made the following request:

SPECIFY LOWER AND UPPER CUTOFF FREQUENCIES (2G15.8)  
1000.0,3000.0

UPPER AND LOWER CUTOFF FREQUENCIES ARE 1000.0000000 3000.0000000 HZ  
1=YES, 0=NO

1

HANNING WINDOW-NF= 255  
ALPHA= 0.5000000  
PRINT OUT WINDOW VALUES(1=YES,0=NO)  
0

ENTER FUNCTION TYPE:  
EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

EXIT

FUNCTION TYPE IS: "EXIT "  
1=YES, 0=NO  
1

The integer signal file is scaled so that the maximum absolute value is 2047.

AN INTEGER SIGNAL FILE OF 6 BLOCKS HAS BEEN CREATED

INTEGER SIGNAL HAS BEEN SCALED BY A FACTOR OF 10.00506  
STOP --

A parameter file was also created and is listed here.

Parameter file name: DMPSIN.PAR

F 18000.000  
TONE  
1024.  
2  
F 100.00000  
F 600.00000  
100.00000 0.00000000 0.00000000  
T 60.000000  
NONE  
0  
0  
NOISE  
1024.  
1  
UNIFORM  
20.000000 0.00000000 1.0000000  
T 0.00000000  
327 964 31855 2

```

NONE
0
1
1
1.0000000
2 5 255
300.00000      0.00000000
0.00000000
EXIT

```

We now call on ISPARS to display the signal in both the time and frequency domain. The default display is 512 points, representing .0284 seconds. The frequency range of the power spectra is 0-9 khz. The display is shown in Figure 1.

#### EXAMPLE TWO - POLNOM

For this example, a fourth order polynomial is created. The equation is  $(t-20)(t-60)(t-150)(t-350) = t^4 - 580t^3 + 93700t^2 - .48 \times 10^7 t + .63 \times 10^8$

For convenience, the sampling frequency has been set to one sample per second. Also, since the values increase quite rapidly after  $t=350$ , the number of samples has been limited to 375. Only one occurrence of this function will be created and the period has been set to 400 seconds so that just one cycle will be displayed. This terminal session will not include "instruction/verification".

```

ENTER EITHER THE SAMPLING FREQ. ("F"G15.8)
OR THE SAMPLING PERIOD ("T"G15.8)

```

F1.0

```

SAMPLING FREQUENCY IS 1.0000000
SAMPLING PERIOD IS    1.0000000

```

```

ENTER FUNCTION TYPE:
EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

```

POLYNOMIAL

FUNCTION TYPE IS: "POLYN"

```

ENTER # OF SAMPLES FOR FIRST OCCURRENCE (F10.0 <CR>
AND # OF OCCURRENCES                      15 <CR>)

```

375  
1

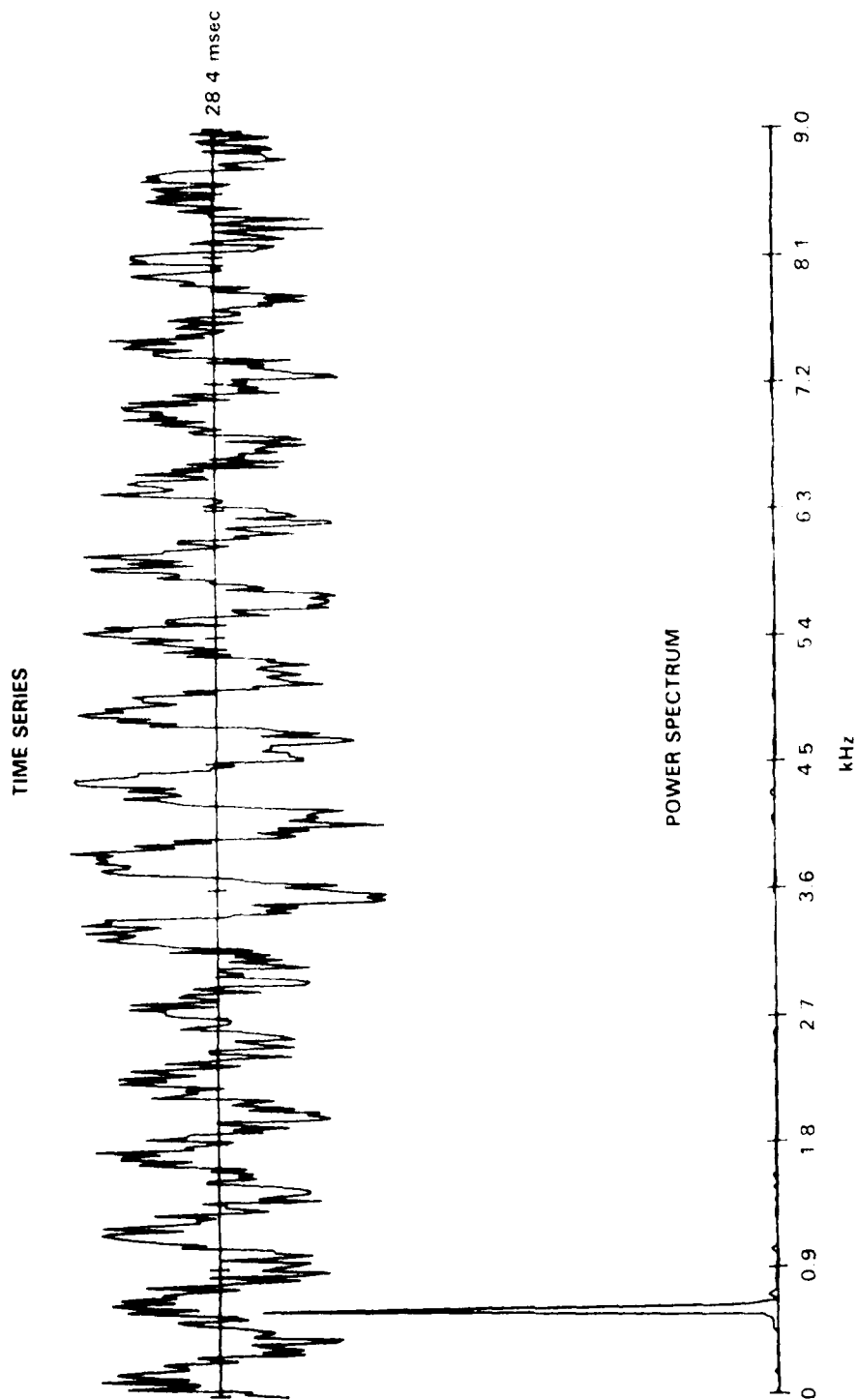


Figure 1 - Time Series and Power Spectrum of DMPSIN

# OF SAMPLES IS 375.  
# OF OCCURRENCES IS 1

ENTER THE POLYNOMIAL FREQUENCY ("F"G15.8) HZ  
OR THE POLYNOMIAL PERIOD ("T"G15.8)  
T400.0

POLYNOMIAL FREQ IS 0.24999999E-02  
POLYNOMIAL PERIOD IS 400.00000  
2 \* PI \* FREQ IS 0.15707962E-01

POLYNOMIAL DUTY CYCLE - ENTER "ON" TIME EITHER AS:  
A FRACTION OF THE PERIOD, UP TO 1.0 ("F"X.XXXXX)  
OR A LENGTH OF TIME, UP TO 400.00000 SECS. ("T"G15.8)  
F1.0

POLYNOMIAL FRAC IS 1.0000000  
POLYNOMIAL TIME IS 400.00000

ENTER POLYNOMIAL ORDER, [0 - 20] (I2) <CR>  
4

ORDER IS 4

Note here that the input format is 5G15.8, which allows up to five values to be entered per line. These values may be entered either in standard floating point format or in scientific notation. Of course, if no decimal point is included, the number is right justified. For example, 12345678 is interpreted to be .12345678

ENTER 5 POLYNOMIAL COEFFICIENTS, ZERO ORDER IST  
IN 5G15.8 FORMAT

.63E8,-.48E7,93700.0,-580.,1.0

POLYNOMIAL COEFFICIENTS ARE:  
63000000. -4800000.0 93700.000 -580.00000  
1.0000000

ENTER OVERALL AMPLITUDE (G15.8)  
1.

AMPLITUDE IS: 1.0000000

For this example, the duty cycle has been set to 1.0 (100%) so that an off-cycle offset is not applicable. If this were not the case, the signal would have a value of (relative amplitude) a(0) + offset during the off-cycle.

ENTER OFFSET FROM 0.0,  
AND FRACTIONAL PHASE (G15.8, P.PPPPP <CR>)  
0.0,0.0

OFFSET, AND FRACTIONAL PHASE ARE:  
0.00000000 0.00000000

Note that the relative amplitude may be modified by a damping factor and  
that if a polynomial is damped, it will asymptotically approach the offset.

ENTER EITHER THE TIME CONST. ("T"G15.8)  
OR THE BANDWIDTH [TC/PI] ("F"G15.8)  
TO.0

TIME CONST. IS 0.00000000  
BANDWIDTH [TC/PI] IS 0.00000000

ENTER MODULATION TYPE:  
AMPLITUDE, FREQUENCY (SPECTRAL), PHASE, NONE, GATEWIDTH

NONE

MODULATION TYPE IS "NONE "

DO YOU WISH TO FILTER THIS SIGNAL ?  
1=YES, 0=NO  
0

DO YOU WISH TO FILTER THE SUMMED SIGNAL ?  
1=YES, 0=NO  
0

ENTER FUNCTION TYPE:  
EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

EXIT

FUNCTION TYPE IS: "EXIT "

AN INTEGER SIGNAL FILE OF 3 BLOCKS HAS BEEN CREATED

INTEGER SIGNAL HAS BEEN SCALED BY A FACTOR OF 0.3254291E-05  
TO A MAX OF + OR - 2047  
STOP --

The parameter file (POLNOM.PAR) is then listed.

F 1.0000000  
POLYNOMIAL  
375.  
F 0.24999999E-02  
F 1.0000000  
4

```

        63000000.    -4800000.0    93700.000    -580.00000    1.0000000
        1.0000000
        0.00000000    0.00000000
T 0.00000000
NONE
0
0
EXIT

```

An X-Y plot from ISPARS is presented in Figure 2.

### EXAMPLE THREE - SINMOD

For this example we will create a 3-khz tone amplitude modulated by a 100-hz tone. "INSTRUCTION/VERIFICATION" was requested.

```

ENTER EITHER THE SAMPLING FREQ. ("F"G15.8)
OR THE SAMPLING PERIOD ("T"G15.8)

```

F18000.0

```

SAMPLING FREQUENCY IS 18000.000
SAMPLING PERIOD IS 0.55555556-04
1=YES, 0=NO
1

```

```

ENTER FUNCTION TYPE:
EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

```

TONE

```

FUNCTION TYPE IS: "TONE "
1=YES, 0=NO
1

```

```

ENTER # OF SAMPLES FOR FIRST OCCURRENCE (F10.0 <CR>
AND # OF OCCURRENCES I5 <CR>)

```

512  
1

```

# OF SAMPLES IS 512.
# OF OCCURRENCES IS 1
1=YES, 0=NO
1

```

Here, only one occurrence is requested, so there is no signal to delay.

UNMOD. VAL = A1 SIN(2 PI FRQ1 + PH1) + M1

```

ENTER EITHER THE TONE FREQUENCY ("F"G15.8) HZ
OR THE TONE PERIOD ("T"G15.8)

```

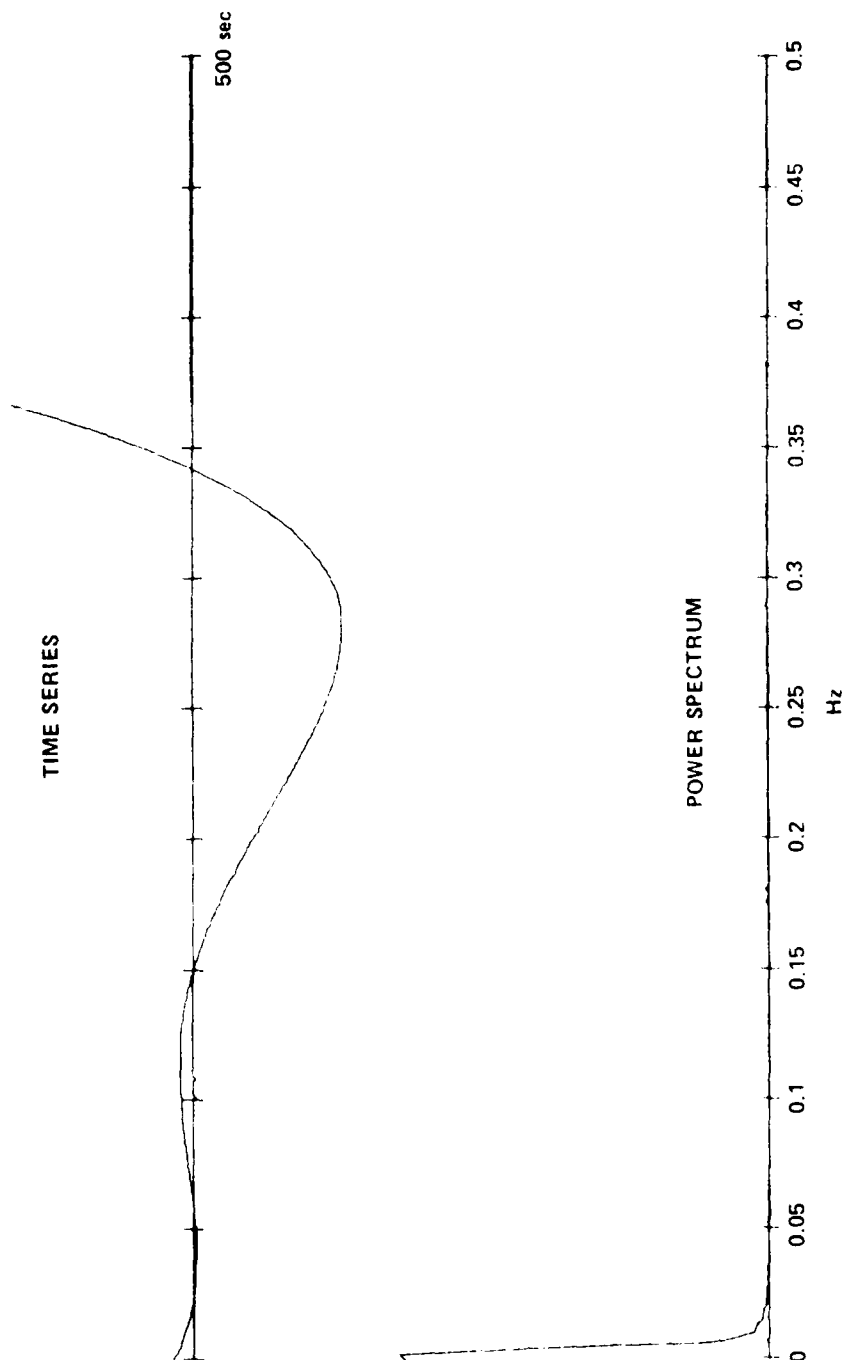


Figure 2 - Time Series and Power Spectrum of POLNOM

F3000.0

TONE FREQ IS 3000.0000  
TONE PERIOD IS 0.33333333E-03  
2 \* PI \* FREQ IS 18849.555  
1=YES, 0=NO  
1

ENTER TONE AMPLITUDE, MEAN,  
AND FRACTIONAL PHASE IN THE FORM:  
G15.8, G15.8, P.PPPPP <CR>

100.0,0.0,0.0

AMPLITUDE, MEAN, AND FRACTIONAL PHASE ARE:  
100.00000 0.00000000 0.00000000  
1=YES, 0=NO  
1

ENTER EITHER THE TIME CONST. ("T"G15.8)  
OR THE BANDWIDTH [TC/PI] ("F"G15.8)

TO.0

TIME CONST. IS 0.00000000  
BANDWIDTH [TC/PI] IS 0.00000000  
1=YES, 0=NO  
1

ENTER MODULATION TYPE:  
AMPLITUDE, FREQUENCY (SPECTRAL), PHASE, NONE  
AMPLITUDE

MODULATION TYPE IS "AMPLITUDE "  
1=YES, 0=NO  
1

ENTER MODULATION FUNCTION:  
TONE, NOISE, GATE, DATA, POLY

TONE

MOD FUNCTION IS: "TONE "  
1=YES, 0=NO  
1

AM VAL =  
A1 [A2 SIN(2 PI FRQ2 + PH2) + M2] SIN(2 PI FRQ1 + PH1) + M1

SPECTRAL FM VAL +  
A1 SIN(2 PI FRQ1 [A2 SIN(2 PI FRQ2 + PH2) + M2] + PH1) + M1

PHASE MOD VAL =  
A1 SIN(2 PI FRQ1 [A2 SIN(2 PI FRQ2 + PH2) + M2] \* PH1) + M1

ENTER MOD TONE AMPLITUDE, MEAN,  
AND FRACTIONAL PHASE IN THE FORM:  
G15.8, G15.8, P.PPPPP <CR>

.75,1.0,0.0

MODULATION AMPLITUDE, MEAN, AND FRACTIONAL PHASE ARE:  
0.75000000 1.0000000 0.00000000

1=YES, 0=NO

1

ENTER THE MOD TONE FREQUENCY ("F"G15.8) HZ  
OR THE MOD TONE PERIOD ("T"G15.8)

F100.0

MODULATION TONE FREQ IS 100.00000  
MODULATION TONE PERIOD IS 0.99999998E-02  
2 \* PI \* MOD. FREQ IS 628.31848

1=YES, 0=NO

1

ENTER EITHER THE MOD TIME CONST. ("T"G15.8)  
OR THE MOD BANDWIDTH [TC/PI] ("F"G15.8)

T0.0

MOD. TIME CONST. IS 0.00000000  
MOD. BANDWIDTH [TC/PI] IS 0.00000000

1=YES, 0=NO

1

DO YOU WISH TO FILTER THIS SIGNAL ?

1=YES, 0=NO

0

DO YOU WISH TO FILTER THIS SUMMED SIGNAL ?

1=YES, 0=NO

0

ENTER FUNCTION TYPE:

EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

EXIT

FUNCTION TYPE IS: "EXIT "

1=YES, 0=NO

1

AN INTEGER SIGNAL FILE OF 3 BLOCKS HAS BEEN CREATED

INTEGER SIGNAL HAS BEEN SCALED BY A FACTOR OF 13.50998  
TO A MAX OF + OR - 2047  
STOP --

SINMOD.PAR is the parameter file.

```
F 18000.000
TONE
    512.
    1
F 3000.0000
100.00000      0.00000000      0.00000000
T 0.00000000
AMPLITUDE
TONE
    0.75000000      1.0000000      0.00000000
F 100.00000
T 0.00000000
0
0
EXIT
```

An X-Y plot, from ISPARS, of both the time and frequency domains is included in Figure 3.

#### EXAMPLE FOUR - FAZMOD

In example four, we will take a previously created parameter file and modify it. The following parameter file (SINE.PAR) describes a pure, unmodulated 1800-hz tone with a relative amplitude of 100.

```
F 18000.000
TONE
    1024.
    1
F 1800.0000
100.00000      0.00000000      0.00000000
T 0.00000000
NONE
0
0
EXIT
```

This signal is displayed in Figure 4.

Here, we reduce both the relative amplitude (which is automatically scaled to 2047) and the frequency and then phase modulate this signal with a 60-hz tone. The synthesizer will read the original parameter file and allow the user to make a parameter-by-parameter change to get the desired results. This occurs because

- 1) The input parameter file is not the terminal (TT:)
- 2) The user requests program interaction

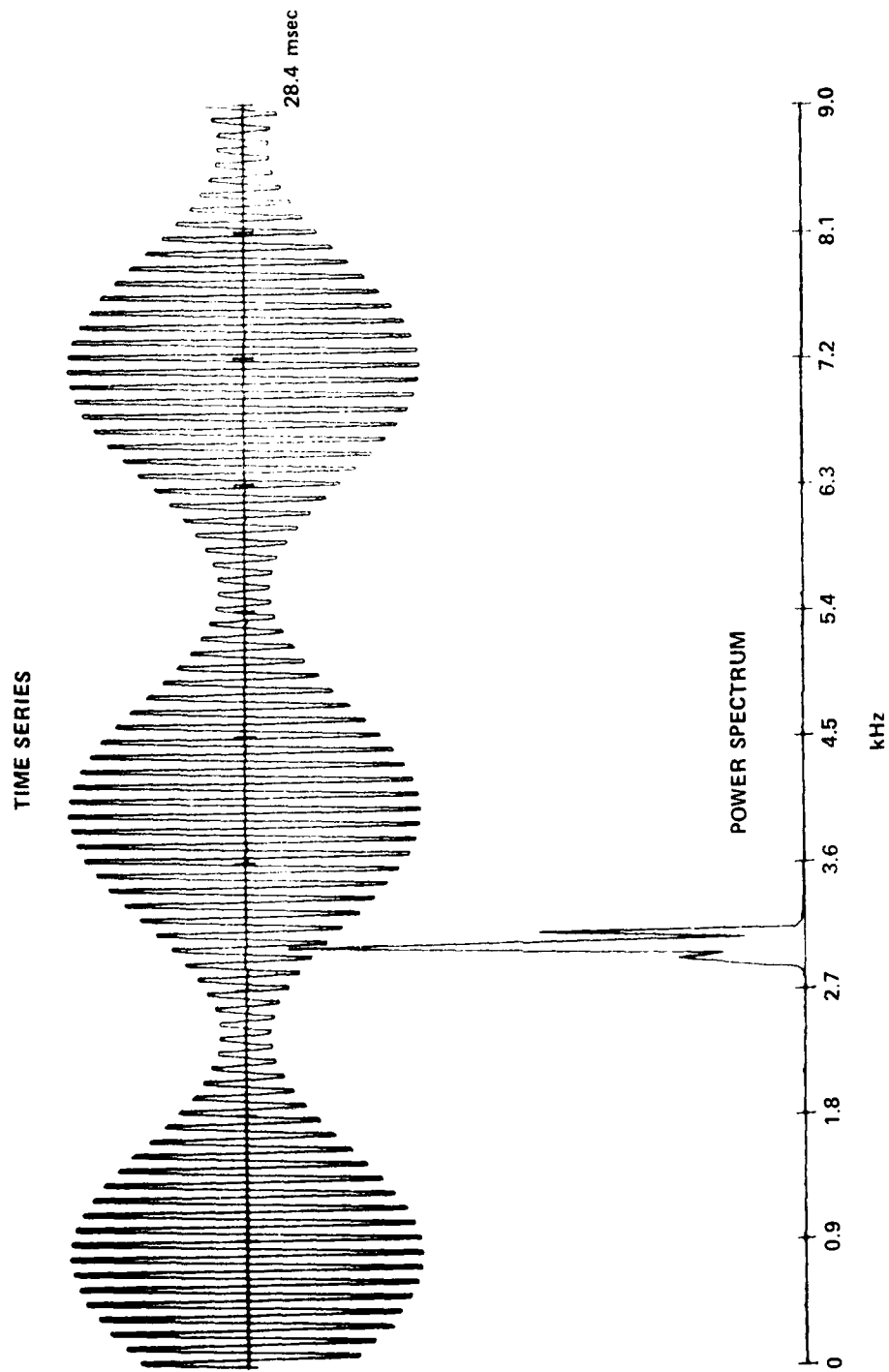


Figure 3 - Time Series and Power Spectrum of SINMOD

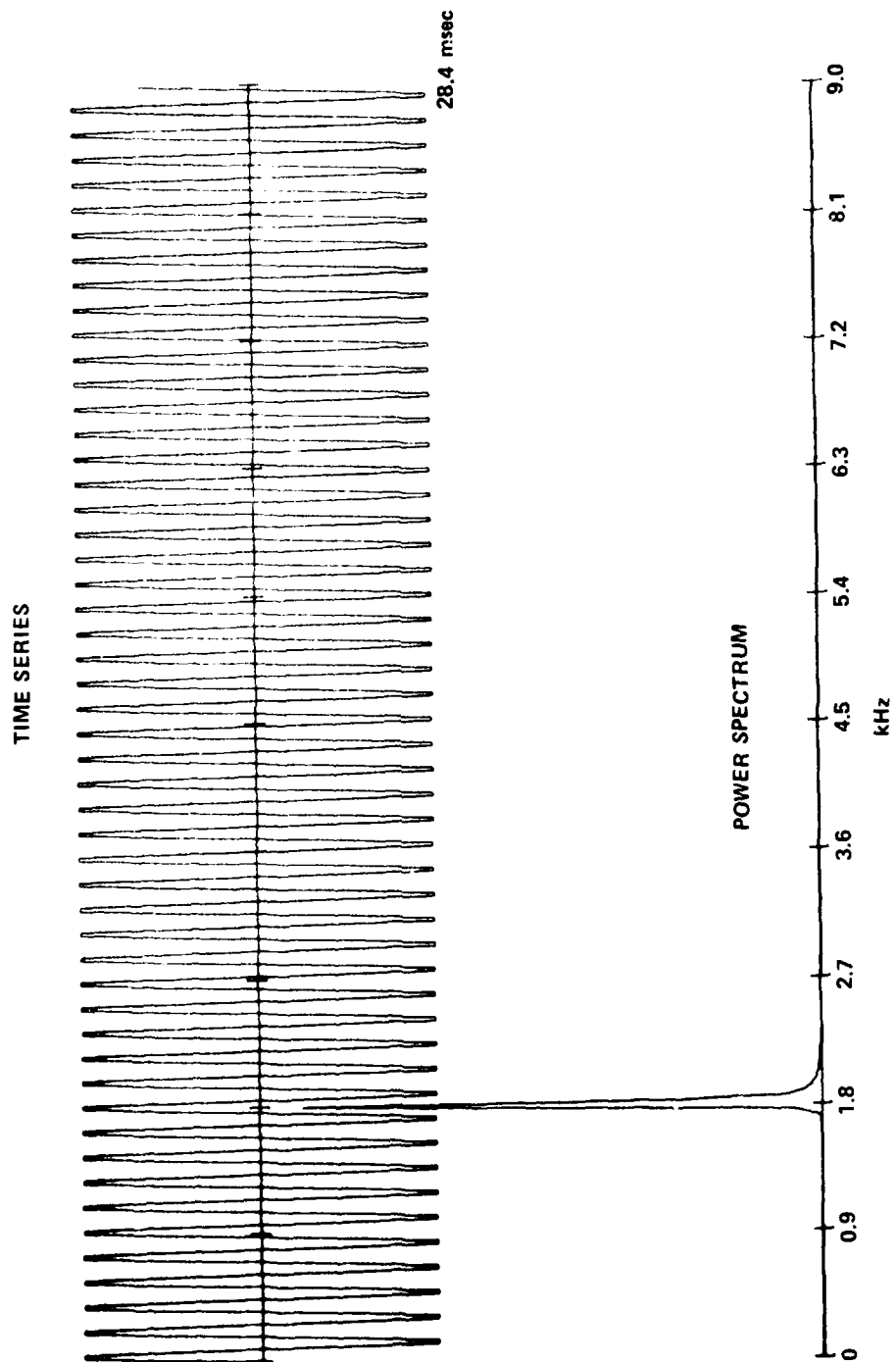


Figure 4 - Time Series and Power Spectrum of SINE

Note that an error condition is created if

- 1) The input parameter file is the terminal
- 2) The user requests no program interaction

In this case, the synthesizer will start over from the beginning.

DO YOU WISH TO INTERACT WITH THE PROGRAM ?

1=YES, 0=NO

1

DO YOU WANT INSTRUCTION / VERIFICATION ?

1=YES, 0=NO

0

ENTER OUTPUT AND INPUT PARAMETER FILE NAMES

\*FAZMOD=SINE

ENTER NAME OF THE INTEGER OUTPUT SIGNAL FILE: "FILE[EXT]="

\*FAZMOD=

SAMPLING FREQUENCY IS 18000.000

SAMPLING PERIOD IS 0.55555556-04

1=YES, 0=NO

1

FUNCTION TYPE IS: "TONE "

1=YES, 0=NO

1

# OF SAMPLES IS 1024.

# OF OCCURRENCES IS 1

1=YES, 0=NO

1

TONE FREQ IS 1800.0000

TONE PERIOD IS 0.55555557E-03

2 \* PI \* FREQ IS 11309.733

1=YES, 0=NO

0

ENTER EITHER THE TONE FREQUENCY ("F"G15.8) HZ  
OR THE TONE PERIOD ("T"G15.8)

F1000.0

TONE FREQ IS 1000.0000

TONE PERIOD IS 0.10000000E-02

2 \* PI \* FREQ IS 6283.1851

1=YES, 0=NO

1

AMPLITUDE, MEAN, AND FRACTIONAL PHASE ARE:  
100.00000 0.00000000 0.00000000  
1=YES, 0=NO  
0

ENTER TONE AMPLITUDE, MEAN,  
AND FRACTIONAL PHASE IN THE FORM:  
G15.8. G15.8, P.PPPPP <CR>  
1.0,0.0,1.0

AMPLITUDE, MEAN, AND FRACTIONAL PHASE ARE:  
1.0000000 0.00000000 1.00000000  
1=YES, 0=NO  
1

TIME CONST. IS 0.00000000  
BANDWIDTH [TC/PI] IS 0.00000000  
1=YES, 0=NO  
1

At this point, we decide to switch from an unmodulated tone to a phase modulated tone.

MODULATION TYPE IS "NONE "  
1=YES, 0=NO  
0

ENTER MODULATION TYPE:  
AMPLITUDE, FREQUENCY (SPECTRAL), PHASE, NONE  
PHASE

MODULATION TYPE IS "PHASE "  
1=YES, 0=NO  
1

The synthesizer temporarily switches operational modes and now accepts input from the terminal until the modulation is fully described.

ENTER MODULATION FUNCTION:  
TONE, NOISE, GATE, DATA, POLY

TONE

MOD FUNCTION IS: "TONE "

ENTER MOD TONE AMPLITUDE, MEAN,  
AND FRACTIONAL PHASE IN THE FORM:  
G15.8, G15.8, P.PPPPP <CR>  
1.0

MODULATION AMPLITUDE, MEAN, AND FRACTIONAL PHASE ARE:  
1.0000000 0.00000000 0.00000000

ENTER THE MOD TONE FREQUENCY ("F"G15.8) HZ  
OR THE MOD TONE PERIOD ("T"G15.8)  
F60.0

MODULATION TONE FREQ IS 60.000000  
MODULATION TONE PERIOD IS 0.16666668E-01  
2 \* PI \* MOD. FREQ IS 376.99109

The user is reminded to enter a carriage return (<CR>) after a PAUSE statement.

ENTER EITHER THE MOD TIME CONST. ("T"G15.8)  
OR THE MOD BANDWIDTH [TC/PI] ("F"G15.8)

1

PAUSE -- ERROR, BAD MOD. TIME CONST. OR BANDWIDTH VALUE  
<CR>

ENTER EITHER THE MOD TIME CONST. ("T"G15.8)  
OR THE MOD BANDWIDTH [TC/PI] ("F"G15.8)

T0.0

MOD. TIME CONST. IS 0.00000000  
MOD. BANDWIDTH [TC/PI] IS 0.00000000

The synthesizer again switches mode of operation and informs the user that the original signal was unfiltered.

FILTER WAS NOT PREVIOUSLY INVOKED AT THIS POINT

DO YOU WISH TO FILTER THIS SIGNAL ?  
1=YES, 0=NO

0

FILTER WAS NOT PREVIOUSLY INVOKED AT THIS POINT

DO YOU WISH TO FILTER THE SUMMED SIGNAL ?  
1=YES, 0=NO

0

FUNCTION TYPE IS: "EXIT "  
1=YES, 0=NO

1

INTEGER SIGNAL FILE 5 BLOCKS HAS BEEN CREATED

INTEGER SIGNAL HAS BEEN SCALED BY A FACTOR OF 2047.069  
STOP --

The new parameter file (FAZMOD.PAR) is then listed.

```

F18000.00000
TONE
  1024.
    1
F 1000.00000
  1.00000      0.00000      1.00000
T   0.00000
PHASE
TONE
  1.00000      0.00000      0.00000
F   60.00000
T   0.00000
0
0
EXIT

```

An X-Y plot of the resultant signal and its power spectra are shown in Figure 5.

#### EXAMPLE FIVE - MODDAT

For the fifth example, we will take a previously created signal as data and modulate it with a ramp function, created by a first order polynomial. The previously created signal is the pure, unmodulated tone used by example four (SIN.PAR) and the parameter file is repeated here for convenience.

```

F 18000.000
TONE
  1024.
    1
F 1800.0000
  100.00000      0.00000000      0.00000000
T 0.00000000
NONE
0
0
EXIT

```

We will also be taking a version of the parameter file for this example and modifying a few of its parameters. The original MODDAT.PAR file is listed here:

```

F 18000.000
DATA
  1024.
    1
  1.0000000      0.000000
T 0.00000000
INTEGER
AMPLITUDE
POLYNOM

```

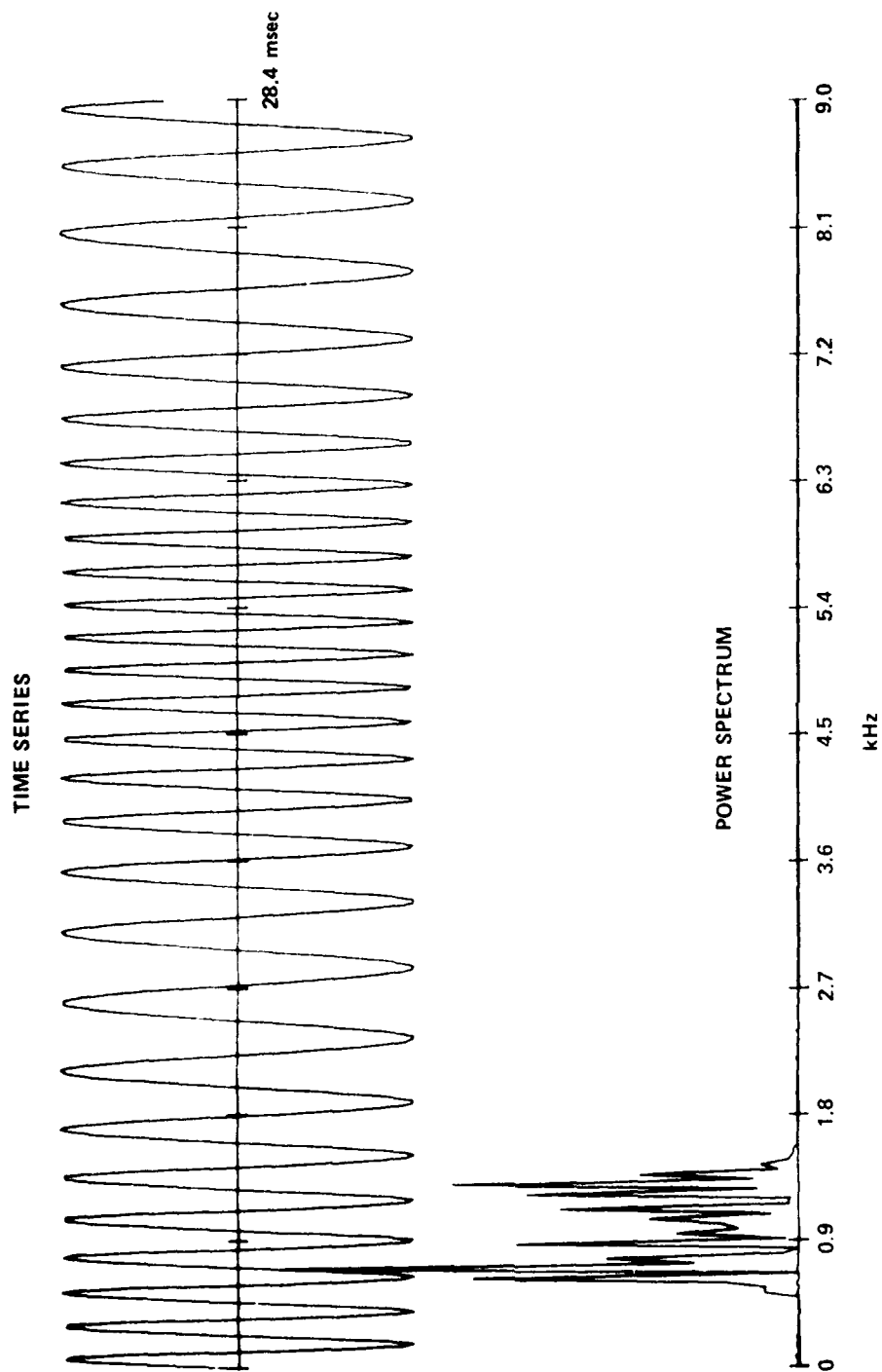


Figure 5 - Time Series and Power Spectrum of FAZMOD

```

F 100.00000
F 1.0000000
1
1.0000000      50.000000
1.0000000
0.0000000      0.0000000
T 0.0000000
0
0
EXIT

```

The interactive session follows:

```

DO YOU WISH TO CREATE AN INTEGER OUTPUT FILE ?
1=YES, 0=NO
1

DO YOU WISH TO CREATE A REAL OUTPUT FILE ?
1=YES, 0=NO
0

DO YOU WISH TO INTERACT WITH THE PROGRAM ?
1=YES, 0=NO
1

DO YOU WANT INSTRUCTION / VERIFICATION ?
1=YES, 0=NO
1

```

The reader should be aware that when the new parameter file name and extension are the same as the old, the old file is destroyed.

ENTER OUTPUT AND INPUT PARAMETER FILE NAMES

IN COMMAND STRING FORMAT:  
 OUTPUT.[EXT]=INPUT.[EXT]

NOTE:

TO INPUT PARAMETERS EXCLUSIVELY FROM THE TERMINAL,  
 SET INPUT PARAMETER FILE NAME TO:  
 "TT:"

.PAR IS DEFAULT EXTENSION

\*MODDAT=MODDAT

ENTER NAME OF THE INTEGER OUTPUT SIGNAL FILE: "FILE[.EXT]="

NOTE:

DEFAULT EXT IS: .SIG

\*MODDAT=

SAMPLING FREQUENCY IS 18000.000  
SAMPLING PERIOD IS 0.55555556E-04  
1=YES, 0=NO

1

FUNCTION TYPE IS: "DATA"  
1=YES, 0=NO

1

# OF SAMPLES IS 1024.  
# OF OCCURRENCES IS 1  
1=YES, 0=NO

1

The synthesizer checks to see whether the data file is in the directory.

ENTER DATA FILE NAME

DEFAULT EXTENSION IS ".DAT"  
\*SIN.SGI

PAUSE -- INPUT FILE DOES NOT EXIST, TRY AGAIN !  
<CR>

ENTER DATA FILE NAME

DEFAULT EXTENSION IS ".DAT"  
\*SINE.SIG

The user has the option to scale, offset, and damp the data.

THE SCALE AND OFFSET ARE: 1.0000000 0.00000000  
1=YES, 0=NO

1

TIME CONST. IS 0.00000000  
BANDWIDTH [TC/PI] IS 0.00000000  
1=YES, 0=NO

1

The present version of the synthesizer allows two data types, real and integer. The reader is reminded that ISPARS will accept only integer data.

DATA TYPE IS "INTEGER "  
1=YES, 0=NO

1

MODULATION TYPE IS "AMPLITUDE "  
1=YES, 0=NO

1

MOD FUNCTION IS: "POLYN"  
1=YES, 0=NO

1

MODULATION

POLYNOMIAL FREQ IS 100.00000  
 POLYNOMIAL PERIOD IS 0.99999998E-02  
 2 \* PI \* FREQ IS 628.31848

1=YES, 0=NO

1

ENTER MODULATION

POLYNOMIAL DUTY CYCLE - ENTER "ON" TIME EITHER AS:

A FRACTION OF THE PERIOD, UP TO 1.0 ("F"X.XXXXX)

OR A LENGTH OF TIME, UP TO 0.99999998E-02 SECS. ("T"G15.8)

F0.7

MODULATION

POLYNOMIAL FRAC IS 0.69999999  
 POLYNOMIAL TIME IS 0.69999998E-02

1=YES, 0=NO

1

MODULATION

ORDER IS 1

1=YES, 0=NO

1

MODULATION

POLYNOMIAL COEFFICIENTS ARE:

1.0000000 50.000000

1=YES, 0=NO

0

Note here that we can selectively change some or all of the previous polynomial coefficients.

ENTER I'TH ORDER [0 - 20] AND ITS COEF. (13, G15.8 <CR>)

ENTER -1 TO FINISH

0,0.0

-1

MODULATION

POLYNOMIAL COEFFICIENTS ARE:

0.00000000 50.000000

1=YES, 0=NO

1

MODULATION

AMPLITUDE IS: 1.0000000

1=YES, 0=NO

1

MODULATION

OFFSET, AND FRACTIONAL PHASE ARE:

0.00000000 0.00000000

1=YES, 0=NO

0

ENTER MODULATION

OFFSET FROM 0.0, AND FRACTIONAL PHASE:

G15.8, P.PPPPP <CR>

0.4,0.0

MODULATION

OFFSET, AND FRACTIONAL PHASE ARE:

0.40000001 0.00000000

1=YES, 0=NO

1

MOD. TIME CONST. IS 0.00000000

MOD. BANDWIDTH [TC/PI] IS 0.00000000

1=YES, 0=NO

1

FILTER WAS NOT PREVIOUSLY INVOKED AT THIS POINT

DO YOU WISH TO FILTER THIS SIGNAL ?

1=YES, 0=NO

0

FILTER WAS NOT PREVIOUSLY INVOKED AT THIS POINT

DO YOU WISH TO FILTER THE SUMMED SIGNAL ?

1=YES, 0=NO

0

FUNCTION TYPE IS: "EXIT "

1=YES, 0=NO

1

AN INTEGER SIGNAL FILE OF 5 BLOCKS HAS BEEN CREATED

INTEGER SIGNAL HAS BEEN SCALED BY A FACTOR OF 1.348973

TO A MAX OF + OR - 2047

STOP --

The new MODDAT.PAR file is

F 18000.000

DATA

1024

1

1.00000000

0.00000000

T 0.00000000

INTEGER

```

AMPLITUDE
POLYNOMIAL
F 100.00000
F 0.69999999
1
0.00000000      50.000000
1.00000000
0.40000001      0.00000000
T 0.00000000
0
0
EXIT

```

For completeness, X-Y plots of the original MODDAT signal, the modulating (ramp) function, and resultant signal are shown in Figures 6, 7, and 8, respectively.

#### EXAMPLE SIX - IMPTRN

For examples six, seven, and eight, we will be using the GATE function to create impulse trains. In example seven, we will use a high-pass filter with a rectangular window and in example eight, we will use a low-pass filter, again with a rectangular window. In both cases, the cutoff frequency is 1 khz.

```

ENTER THE GATE FREQUENCY ("F"G15.8)HZ
OR THE GATE PERIOD ("T"G15.8)
F150.0

```

```

GATE FREQ IS      150.00000
GATE PERIOD IS    0.66666667E-02
2 * PI * FREQ IS  942.47772

```

Since we are not adding any other functions to this one, both the amplitude and offset could just as well have been set to any other positive number.

```

ENTER "MEAN-PK' AMP, MEAN, AND FRACTIONAL PHASE:
G15.8, G15.8, 0.PPPPP <CR>
1.0,1.0,0.0

```

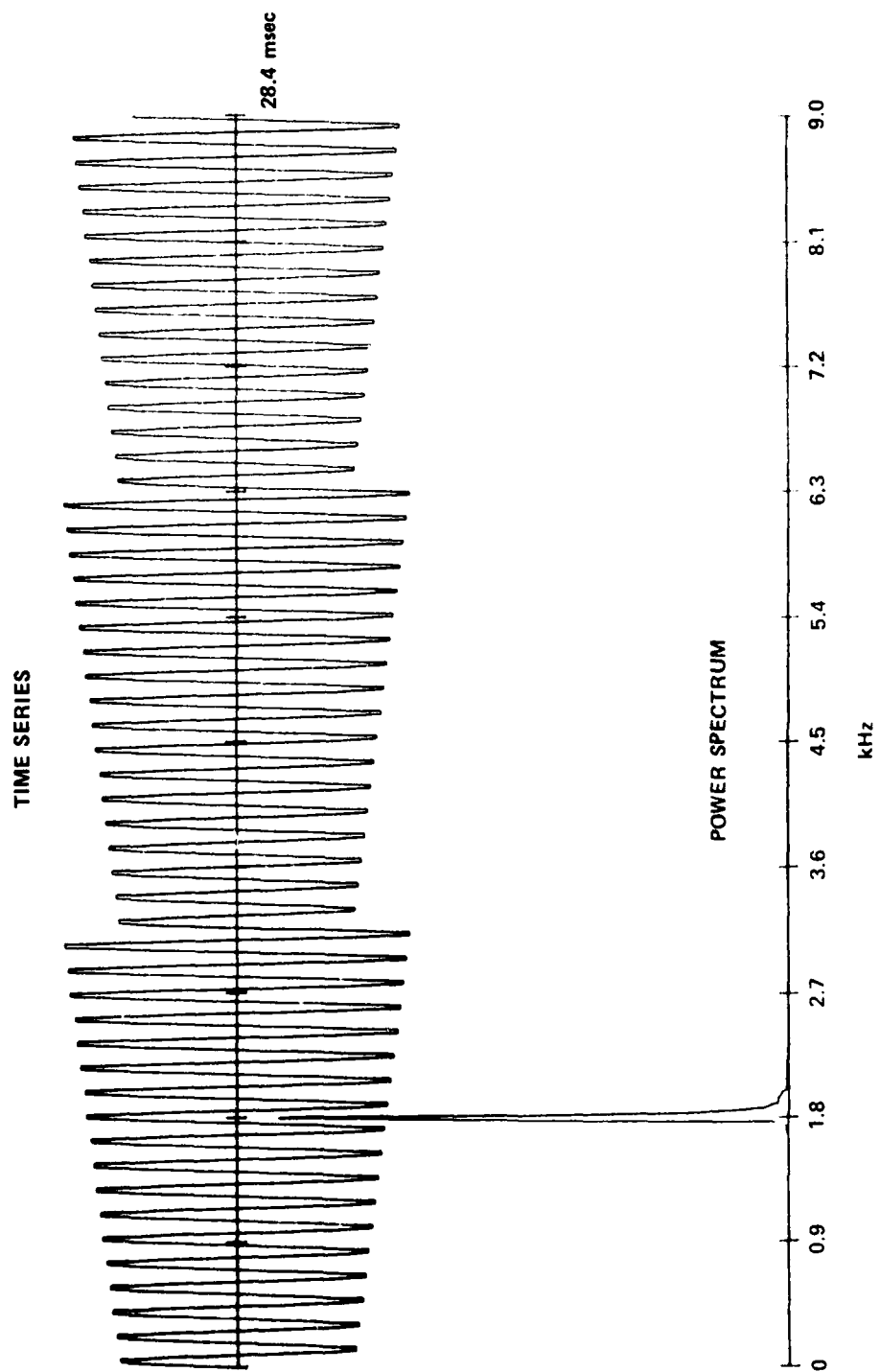


Figure 4 - Time Series and Power Spectrum of Original MODDAI

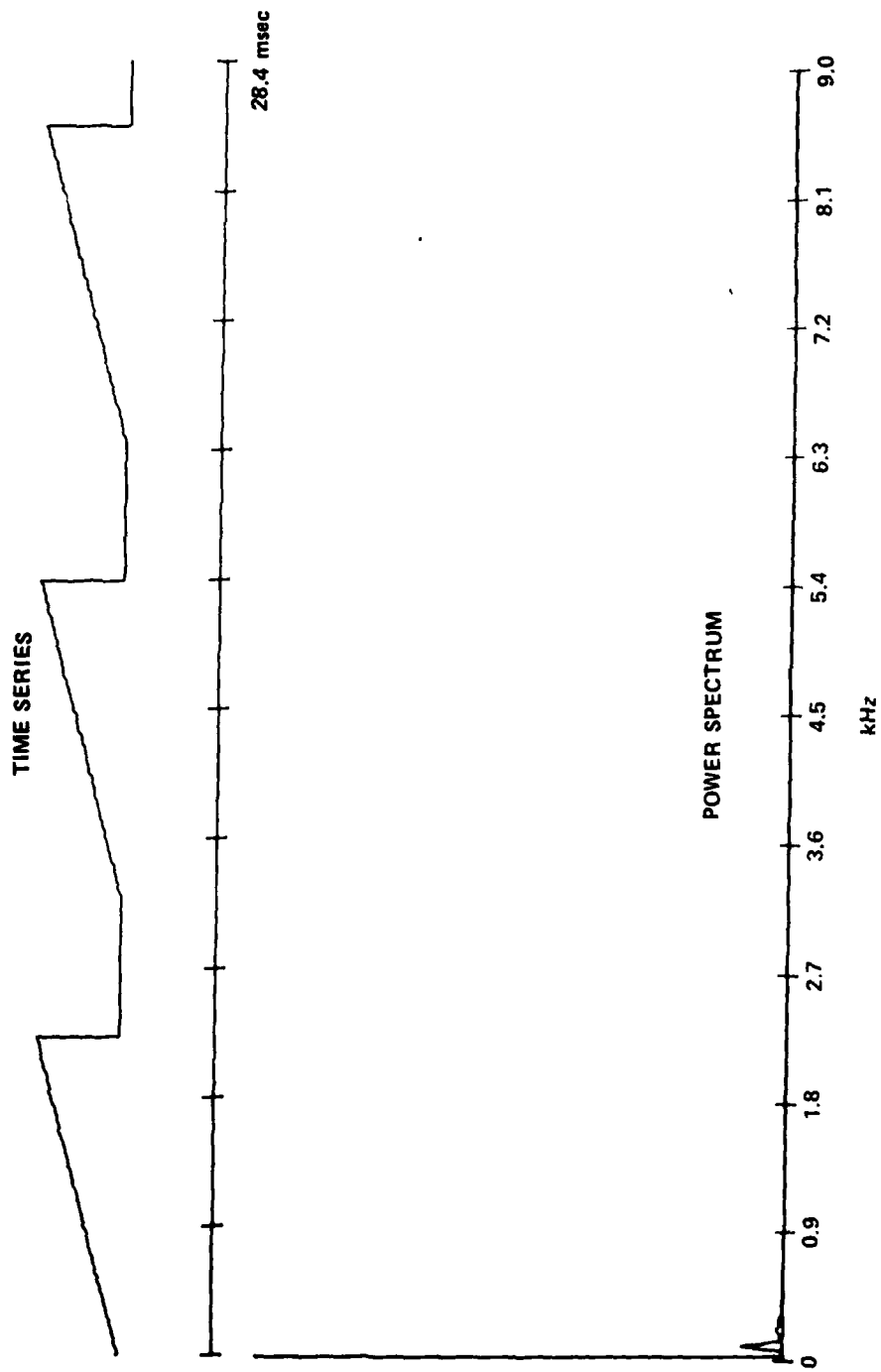


Figure 7 - Time Series and Power Spectrum of RAMP

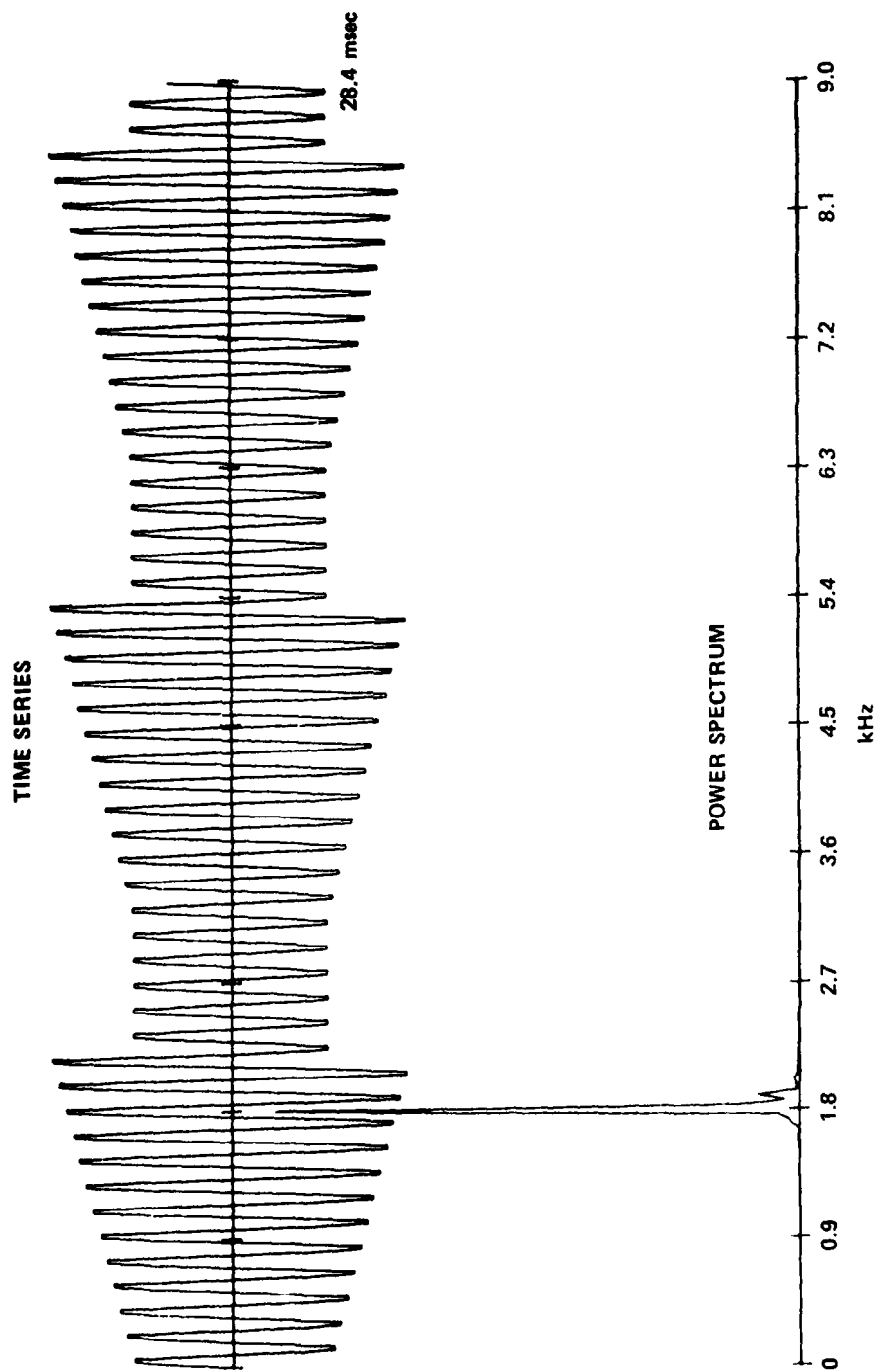


Figure 8 - Time Series and Power Spectrum of MODDAT

AMPLITUDE,            OFFSET,    AND FRACTIONAL PHASE ARE:  
1.0000000            1.0000000            0.0000000

If the user requests a zero "on" time for the duty cycle, the synthesizer recognizes it as a request to create an impulse train and translates that to be one "on" sample per period.

GATE DUTY CYCLE - ENTER "ON" TIME EITHER AS:  
A FRACTION OF THE PERIOD (F"0.XXXXX)  
OR A LENGTH OF TIME, UP TO 0.66666667E-02 SECS. ("T"G15.8)  
"FO.0" OR "TO.0" -> IMPULSE FUNCT  
FO.0

GATE FRAC IS            0.83333334E-02  
GATE TIME IS            0.55555556E-04

ENTER EITHER THE TIME CONST.            ("T"G15.8)  
                                 OR THE BANDWIDTH [TC/PI] ("F"G15.8)  
TO.0

TIME CONST. IS            0.00000000  
BANDWIDTH [TC/PI] IS    0.00000000

ENTER MODULATION TYPE:  
AMPLITUDE, FREQUENCY (SPECTRAL), PHASE, NONE, GATEWIDTH

NONE

MODULATION TYPE IS "NONE        "

DO YOU WISH TO FILTER THIS SIGNAL ?  
1=YES, 0=NO  
0

DO YOU WISH TO FILTER THE SUMMED SIGNAL ?  
1=YES, 0=NO  
0

ENTER FUNCTION TYPE:  
EXIT, TONE, NOISE, GATE, DATA, POLYNOMIAL

EXIT

FUNCTION TYPE IS: "EXIT "

AN INTEGER SIGNAL FILE OF        5 BLOCKS HAS BEEN CREATED

INTEGER SIGNAL HAS BEEN SCALED BY A FACTOR OF    1023.500  
TO A MAX OF + OR - 2047  
STOP --

The parameter file (IMPTRN.PAR) is listed here.

```

F 18000.000
GATE
    1024.
    1
F 150.00000
  1.0000000      1.0000000      0.0000000
F 0.00000000
T 0.00000000
NONE
0
0
EXIT

```

The X-Y plot from ISPARS is included in Figure 9.

For the remaining three examples, we will give only the parameter files and the X-Y plots of both the signal and its power spectra.

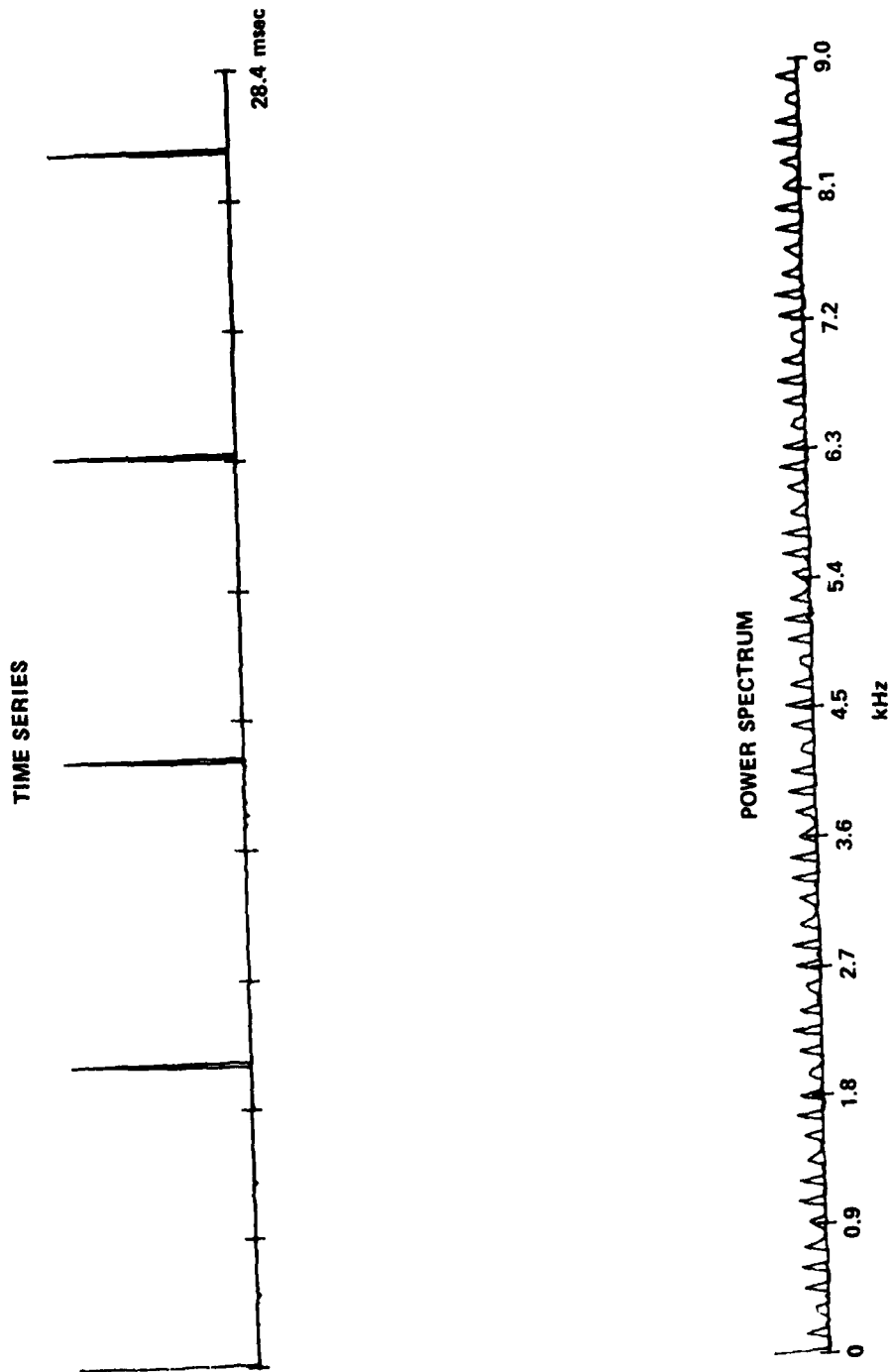


Figure 9 - Time Series and Power Spectrum of IMPTRN

# EXAMPLE SEVEN - IMPTRH

```

F 18000.000
GATE
  1024.
  1
F 150.00000
  1.0000000      1.0000000      0.0000000
F 0.0000000
T 0.0000000
NONE
1
  1
    1.0000000
  2 1 255
    1000.0000      0.0000000
    0.0000000
0
EXIT

```

X-Y plots of IMPTRH and IMPTRL are shown in both the time and frequency domains in Figures 10 and 11 respectively.

# EXAMPLE EIGHT - IMPTRL

```

F 18000.000
GATE
  1024.
  1
F 150.00000
  1.0000000      1.0000000      0.0000000
F 0.0000000
T 0.0000000
NONE
1
  1
    1.0000000
  1 1 255
    1000.0000      0.0000000
    0.0000000
0
EXIT

```

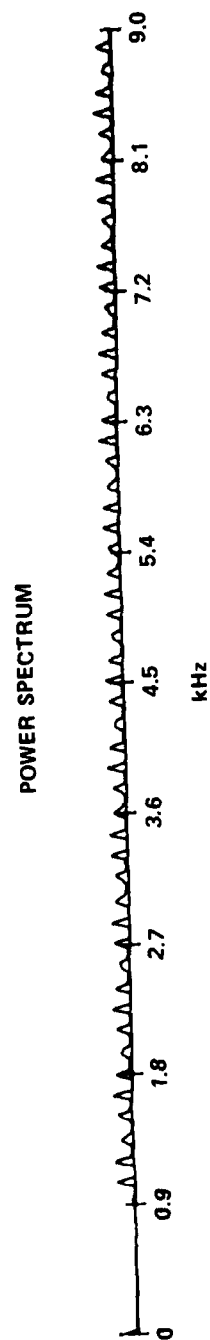
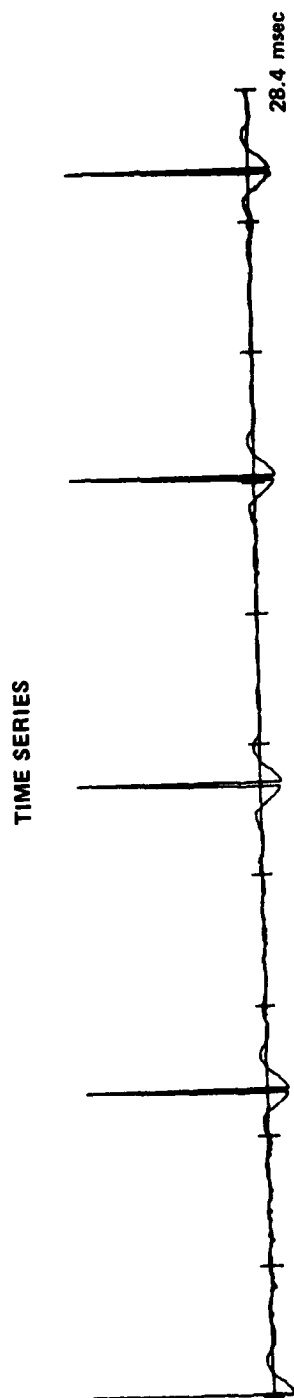


Figure 10 - Time Series and Power Spectrum of IMPTRH

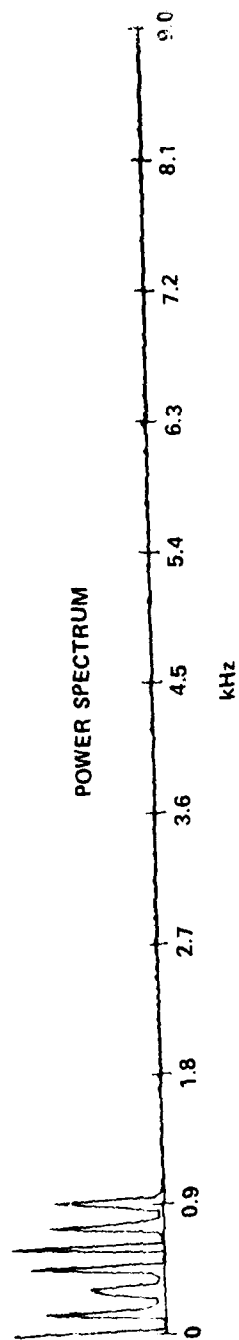
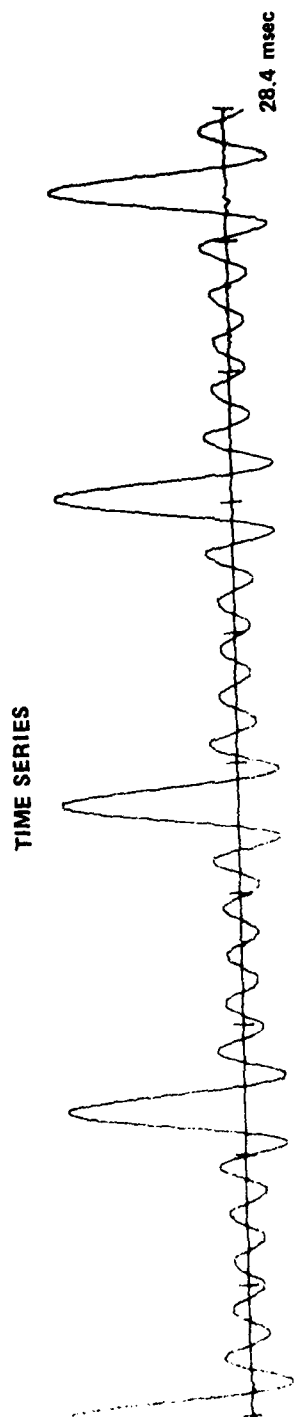


Figure 11 - Time Series and Power Spectrum of IMPTRL

# EXAMPLE NINE - NOISGT

```

F 18000.000
GATE
  1024.
  1
F 100.00000
  100.00000      0.00000000      0.00000000
F 0.50000000
T 0.00000000
NONE
0
0
NOISE
  1024.
  1
GAUSSIAN
  40.000000      0.00000000      2.00000000
T 0.00000000
  32 8654      3 22222
NONE
0
0
EXIT

```

The X-Y plot of NOISGT is shown in both the time and frequency domains in Figure 12.

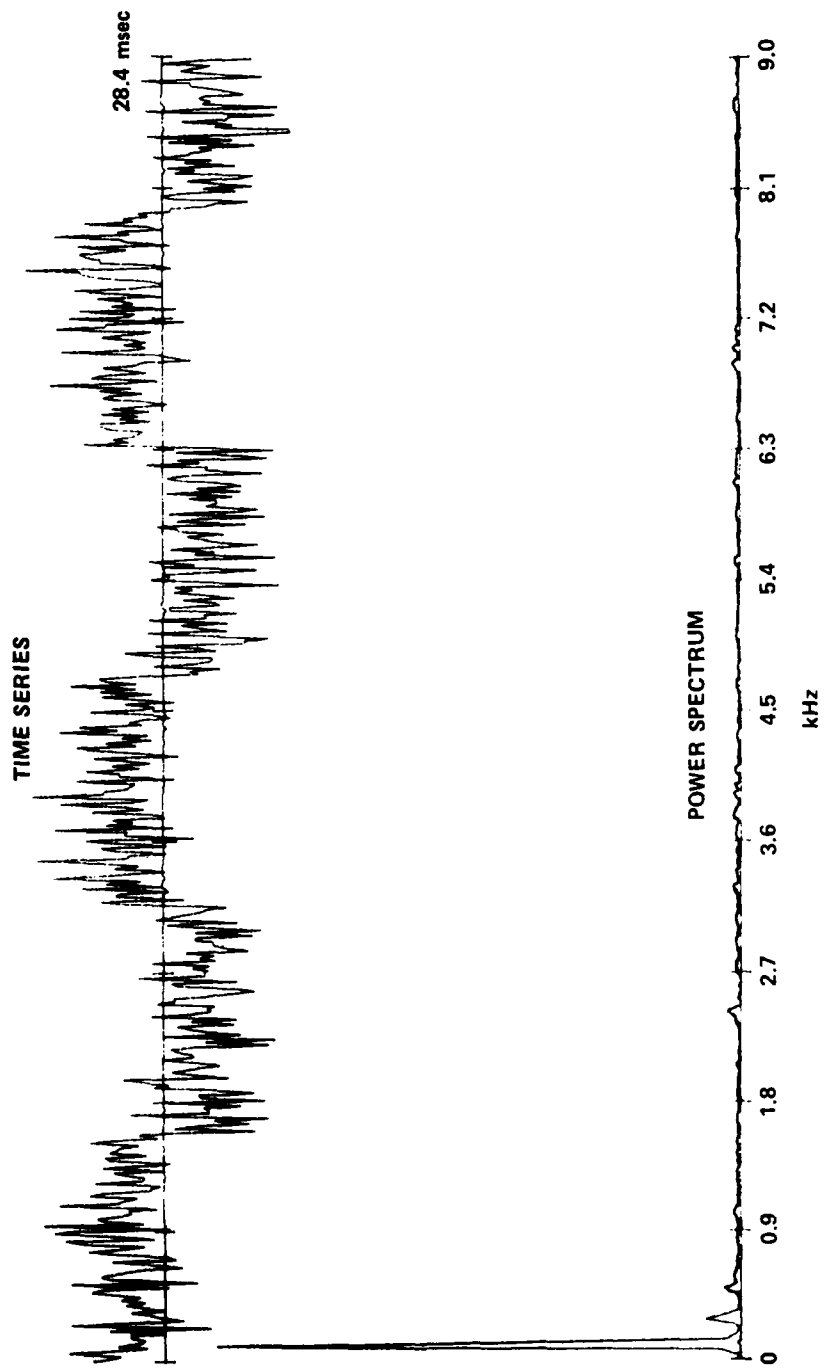


Figure 12 - Time Series and Power Spectrum of NOISGT

#### APPENDIX - DEFAULT EXTENSION NAMES

PAR	Input/Output	- Parameter file
SIG	Output	- Integer signal file
WAV	Output	- Real signal file
DAT	Input	- Data and modulating data files

## REFERENCES

1. Berkowitz, S., "Functional Description of an Interactive Signal Synthesis System," DTNSRDC Report 82/073 (Sep 1982).
2. Parsons, W. et al., "ISPARS - Interactive Signal Pattern Analysis and Recognition System," DTNSRDC Report in publication.
3. "RT-11 System Reference Manual," Digital Equipment Corporation, Maynard, Mass. (1974).

PREVIOUS PAGE  
IS BLANK

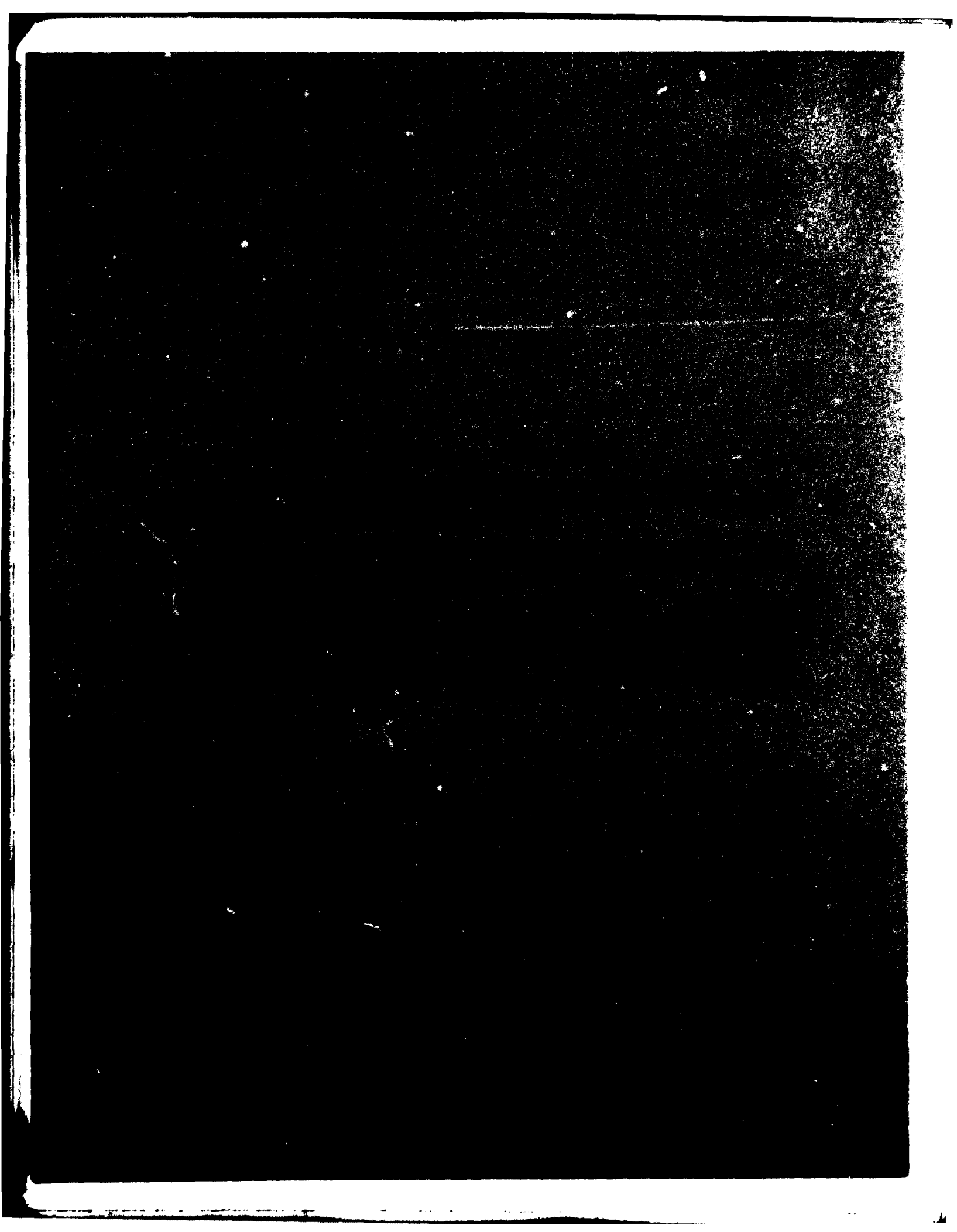
# INITIAL DISTRIBUTION

Copies		Copies	Code	Name
1	CNR, 220/T. Warfield	1	28	J. Belt
1	DNL	10	5211.1	Reports Distribution
1	NAVMAT	1	522.1	TIC (C)
1	NRL	1	522.2	TIC (A)
2	NAVSEA	1	93	L. Marsh
	1 03F/J. Sejd			
	1 55N/L. Slavin			
1	NUSC			
12	DTIC			

## CENTER DISTRIBUTION

Copies	Code	Name
1	15	W. Morgan
1	16	H. Chaplin
1	17	W. Murray
1	18	G. Gleissner
2	1808	D. Wildy
1	182	A. Camara
1	1824	S. Berkowitz
3	1824	I. Zaritsky
1	184	J. Schot
1	185	T. Corin
1	187	M. Zubkoff
1	19	M. Sevik
1	1925	G. Clark
1	193	J. O'Donnell
1	27	W. Dietz

PREVIOUS PAGE  
IS BLANK



END

DATE  
FILMED

4 - 84

DTIC